UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Universitat Politècnica de València

Escola Tècnica Superior d'Enginyeria Informàtica

Proyecto Final de Carrera

"Desarrollo de una aplicación básica en SAKAI"

Código: DISCA-312

Titulación: ITIS

Autores:

JOSEP ADRIÀ SANSALONI MELIS

JOSE MONTAGUT VIDAL

Dirigido por:

Dr. LENIN LEMUS ZUÑIGA

PROFESOR TITULAR

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Septiembre 2012

Índice

Índice de Figuras

Índice de listados de código

INTRODUCCIÓN

En este proyecto vamos a trabajar con la plataforma Sakai, explicaremos su instalación paso a paso en el sistema operativo Linux, concretamente la distribución de Ubuntu, crearemos un script para su automatización y desarrollaremos una aplicación en su entorno.

Sakai es un sistema para la enseñanza, el aprendizaje, la investigación y la colaboración mediante la tecnología con funcionalidad total. Para poder instalar su plataforma de trabajo vamos a necesitar los siguientes componentes:

- JDK de Java versión 1.7
- MySQL Server
- Maven2
- Subversión
- Tomcat
- Conector de MySQL
- Eclipse

En el siguiente capítulo vamos a introducir los componentes más importantes para entender su funcionamiento y poder trabajar con ellos.

Este proyecto ha sido dessarrollado por:

- Josep Adria Sansaloni Melis
- Jose Montagut Vidal

Los Capitulos que se han dessarrollado en grupo son todos excepto el capitulo seis y el siete.

El capitulo seis ha sido desarrollado por Josep Adria y el siete ha sido desarrollado por Jose.

OBJETIVO

El objetivo principal de este proyecto:

“Crear un script que permita crear café con el fin de verificar que una aplicación SAKAI se ejecuta de forma apropiada en “SAKAI Programmer’s Café”

Para conseguir este objetivo instalamos la plataforma de Sakai y utilizaremos el framework wicket para trabajar de forma sencilla con el lenguaje J2EE que es con el que trabaja Sakai.

Como objetivo secundario queremos automatizar la instalación de la plataforma Sakai mediante unos scripts, los cuales haremos referencia en capítulos posteriores

METODOLOGÍA A SEGUIR

1. Documentarnos el método de instalación de la plataforma de Sakai.
2. Obtener información sobre todos los elementos y su funcionamiento necesarios para la correcta instalación de la plataforma Sakai.
3. Realización de un análisis en los Sistemas Operativos más adecuados para la instalación de la plataforma.
4. Selección del Sistema Operativo que mejor se adapta al objetivo del proyecto
5. Instalación de la plataforma Sakai.
6. Instalación del entorno de desarrollo de aplicaciones para Sakai.
7. Desarrollo del script de instalación.
8. Desarrollo de la Aplicación.
9. Testear el correcto funcionamiento de los archivos desarrollados.

CAPITULO 1. DEFINICIONES BÁSICAS

Como se ha comentado en el capítulo anterior vamos a explicar los componentes más importantes para la instalación de Sakai.

1.1. MySQL

MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalista para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MyQSL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. Esta puede ser desde una simple lista de compras a toda la información de una red corporativa. Para agregar, acceder y procesar todos los datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. MySQL utiliza SQL (Structured Query Language) como formato para hacer las consultas, este es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y que no puede hacer con el software en diferentes situaciones.

MySQL Database Server es muy rápido, confiable y fácil de usar, tiene un práctico set de características desarrollado en cercana cooperación con nuestros usuarios. MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. MySQL Server ofrece hoy una rica variedad de funciones, y su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para acceder a bases de datos en Internet.

El software de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL multi-threaded que trabaja con diferentes bakends, programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones.

La estructura general de las consultas a base de datos es:

```
SELECT [ALL|DISTINCT] lista_items_seleccionados | *
FROM lista_tablas
[WHERE expresión_condicional]
[GROUP BY lista_columnas]
[HAVING expresión_condicional]
[ORDER BY lista_columnas]
```

Sintaxis general del lenguaje:

- Cadenas de caracteres deben limitarse por " o '.
- Tiene soporte completo para las cláusulas SQL GROUP BY y ORDER BY se puede especificar ASC y DESC.
- Tiene soporte de funciones: COUNT(), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP_CONCAT().
- Tiene soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN.
- Tiene soporte para poner alias en tablas y columnas como lo requiere el estándar SQL.
- Las sentencias DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado.
- El comando específico SHOW se utiliza para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices.
- El comando EXPLAIN puede usarse para determinar cómo el optimizador resuelve una consulta.
- Los nombres de funciones no colisionan con los nombres de tabla o columna. La única restricción en la llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Puede mezclar tablas de distintas bases de datos en la misma consulta `db_nombre.tbl_nombre`.
- Las variables se pueden inicializar en un comando con el operador :=
- MySQL entiende los operadores || y && como OR y AND lógicos.
- El operador % es sinónimo de MOD()
- Los operadores =, <>, <= ,<, >=,>, <<, >>, <=>, AND, OR, o LIKE se utilizan en expresiones condicionales, esto quiere decir, que se utiliza para comparar las columnas entre sí o con un valor definido

Algunas Características de MySQL:

- Las columnas pueden ser de diversos tipos como enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM.
- Sentencias y funciones:
- Uno de los parámetros más importante es la seguridad, MySQL tiene un sistema de privilegios y contraseñas que es muy flexible y seguro. Este permite la verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
- Sabemos que los límites son hasta 60.000 tablas y cerca de 5.000.000.000.000 de registros. Se permiten hasta 64 índices por tabla. Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El límite máximo de ancho son 1000 bytes. Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.
- Hay muchas formas de conectarse a MySQL que son:
    - Los clientes se conectan usando sockets TCP/IP en cualquier plataforma.
    - En MySQL 5.0 o posterior, los servidores Windows soportan conexiones con memoria compartida.
    - La interfaz para el conector ODBC proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity).
    - La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC.
- MySQL está disponibilidad para una grandísima cantidad de plataformas y Sistemas Operativos.
- Búsqueda e indexación de campos de texto que esto hace que las búsquedas sean muchísimo más rápidas.
- Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución:
    - Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example
    - Desarrollados por partners como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright, Kickfire, XtraDB.
    - Desarrollados por la comunidad como memcache, httpd, PBXT y Revisión.

- MySQL permite Agrupar múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Instalación de MySQL en Ubuntu.

Hay varias formas de instalar el servidor de MySQL en Linux, explicare la forma más fácil de instalar que es mediante administrador de paquetes nativo de Linux. En este tipo de distribuciones similares a debían hay dos paquetes mysql-client and mysql-server. Me centrare en la instalación del mysql-server que es el que utilizaremos para Sakai. Para instalar simplemente tiene que escribir este comando en la consola de Linux y se instalará todo automáticamente.

```
sudo apt-get install mysql-server-x.x
```

Donde x.x es la versión de MySQL que vas a instalar.

Durante la instalación la base se creará y le pedirá la contraseña administradora de MySQL y su confirmación. Se creara un archivo de configuración en /etc/mysql/my.cnf y un script /etc/init.d/mysql que inicializará el servició automáticamente cuando se encienda el sistema operativo.

Comandos que podemos usar en la consola de Unix.

- mysql -u [username] –p Sirve para entrar dentro de MySQL.
- mysql -u [username] -p [database name] sirve para entrar MySQL y selecciona la base de datos en la que entra.
- use [database name]; Selecciona la base de datos que quieres usar.
- show databases; Muestra las base de datos que hay dentro de MySQL.
- show tables; Muestra las tablas que hay dentro de la base de datos seleccionada.
- CREATE TABLE nombre_table (nombre_variable tipo, nombre_variable tipo,…); Crea la tabla en la base de datos seleccionada con las columnas que están entre paréntesis.
- select * from my _table; Muestra todo lo que hay dentro de la table my_table.
- quit; para salir de MySQL y regresar a la consola.

1.2. MAVEN

Maven es una herramienta de software que se utiliza para la gestión y producción de proyectos Java. Maven es una palabra que proviene del Yiddish y tiene el significado de alguien experto, o de un acumulador de conocimientos. Esta herramienta de software fue creada por Jason van Zyl, de Sonatype, en 2002. Maven apareció por la necesidad de los desarrolladores de tener una manera estándar de construir los proyectos, una definición clara sobre lo que un proyecto consistía, una forma fácil de publicar la información del proyecto y una forma de compartir archivos JAR a través de varios proyectos.

Las principales características de Maven son:

- Estandariza la estructura de directorios: Maven propone una misma estructura para todos los proyectos, pero permitiendo que pueda configurarse de forma personalizada en caso necesario.
- Estandariza el ciclo de vida: Maven aporta una implementación de ciclo de vida. De modo que ejecutar cualquier fase del ciclo de vida es siempre igual en todos los proyectos Maven. Para el desarrollador común aprender a manejar un proyecto Maven implica haber aprendido a manejar todos los proyectos Maven. Además este ciclo de vida es extensible permitiendo añadir tareas personalizadas.
- Reutilización: Al utilizar Maven implantar otros proyectos es algo instantáneo porque es reutilizable.
- Integración: Maven está integrado con la mayoría de herramientas y frameworks que se utilizan en la actualidad, algunos ejemplos serían Eclipse, Selenium, Subversion...
- Gestión de dependencias: Maven aporta un sistema de gestión de dependencias basado en repositorios y una fuerte configuración en el proyecto.

Los comandos más habituales que se suelen utilizar en Maven son:

- mvn clean: Limpia el directorio target del proyecto. En este directorio se guardan los archivos compilados y todos los recursos generados por el proyecto.
- mvn compile: Compila el código fuente del proyecto.
- mvn test: Ejecuta los tests unitarios del proyecto.
- mvn package: Empaqueta nuestro proyecto en un archivo jar, war, ear, etc dependiendo de la configuración de nuestro proyecto.
- mvn install: Guarda en nuestro repositorio local los "artefactos" generados por el comando package.
- mvn deploy: Guarda en el repositorio remoto, configurado en el proyecto, los "artefactos" generados por el comando package.
- mvn jetty-run: Arranca un servidor de aplicaciones (Jetty) donde podemos probar nuestros proyectos j2ee. La ruta en la cual estarán disponibles será http://localhost:8080.

- mvn hibernate3:hbm2ddl: Genera o regenera el esquema de base de datos, consultando el fichero de configuración de hibernate.
- mvn dbunit:operation: Sube los datos contenidos en el fichero sample-data.xml a la base de datos.
- mvn dbunit:export: Exporta todos los datos de la bd a un fichero xml que podremos encontrar en el directorio target del proyecto.
- mvn archetype:create -DgroupId=org.yaxche.blog.ejemplos -artifactId=blog.ejemplos.maven.comandosutiles : Este comando crea un proyecto.
- mvn archetype:create -DgroupId=org.yaxche.blog.ejemplos -DartifactId=blog.ejemplos.maven.comandosutiles.web -DarchetypeArtifactId=maven-archetype-webapp : Este comando es similar al anterior porque deriva de él ya que se utiliza para crear un proyecto web.

Maven tiene dos versiones Maven 1 y Maven 2.En la primera versión Maven puede mediante un fichero XML y una serie de extensiones (plugins) compilar el proyecto java, ejecutar una serie de pruebas unitarias, generar paquetes y generar una serie de informes. En la segunda versión también utiliza un fichero XML denominado pom.xml (Project Object Model) y puede hacer lo mismo que en la versión 1 pero con la diferencia de que puede gestionar de manera automática las dependencias de los proyectos que gestiona.

1.3. POM

POM es la abreviación de modelo de proyecto de objeto (Project Object Model) y es la unidad de trabajo fundamental de Maven. Es un archivo XML que contiene información sobre los proyectos y detalles de configuración utilizados por Maven para construir los proyectos. Contiene los valores por defecto para la mayoría de los proyectos.

El POM fue renombrado de project.xml en Maven 1 al pom.xml de Maven 2. En lugar de tener un archivo XML que contiene los objetivos que se pueden ejecutar, los objetivos o los plugins se configuran ahora en el pom.xml. Al ejecutar una tarea u objetivo, Maven busca el POM en el directorio actual. Lee el POM, obtiene la información de configuración necesaria, y a continuación, ejecuta el objetivo. Parte de la configuración que se puede especificar en el POM son las dependencias del proyecto, plugins o los objetivos que se pueden ejecutar y los perfiles de construcción. Otra información como la versión del proyecto, descripción y desarrolladores también se puede especificar.

Los requisitos mínimos para un POM son los siguientes:

- origen del proyecto.
- modelVersion - Debería ser establecido a 4.0.0.
- groupId - La id del grupo del proyecto.
- artifactId - La id del artefacto(proyecto).
- version - La versión del artefacto en el grupo especificado.

A continuación podemos ver un ejemplo:

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1</version>
</project>
```

Un POM requiere que su groupId, artifactId, y version sean configurados.

1.4. SUBVERSION

Subversion es un sistema de control de versiones de código abierto.

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma

manual, es muy aconsejable disponer de herramientas que faciliten esta gestión y ahí es donde se sitúa el uso de Subversion.

Subversion gestiona ficheros y directorios, y los cambios introducidos en ellos, según el tiempo. Esto te permite recuperar versiones antiguas de tus datos o examinar el historial de como tus datos han cambiado. En este sentido, muchas personas piensan en un sistema de control de versiones como una especie máquina del tiempo. Subversion ayuda a que los desarrolladores lleven un seguimiento de los cambios en los ficheros de código fuente de su proyecto.

Subversion puede acceder a los repositorios a través de redes lo que permite que se pueda acceder al repositorio desde distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada —si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio.

Subversion, una vez instalado, tiene un número de partes diferentes. La siguiente es una descripción rápida de dichas partes:

-svn- La línea de comandos del programa cliente.

-svnversion- Un programa para reportar el estado de una copia de trabajo.

-svnlook- Una herramienta para inspeccionar directamente un repositorio Subversion.

-svnadmin- Una herramienta para crear, modificar o reparar un repositorio Subversion.

-mod_dav_svn- Un módulo plug-in para el Servidor Apache HTTP, utilizado para hacer disponible su repositorio a otros a través de una red.

-svnserve- Un programa de servidor personalizado independiente, ejecutable como proceso demonio o invocarle por SSH. Otra manera de hacer disponible su repositorio a otros a través de una red.

-svndumpfilter- Un programa para filtrar repositorios de Subversion.

-svnsync- Un programa para reflejar de forma incremental un repositorio a otro en una red.

Algunos comandos básicos para utilizar Subversion son:

- *svn import <ruta>/proyecto http://svn.servidor.net/proyecto/trunk -m 'Descripción de la creación'* -->Importación inicial del proyecto al repositorio.
- *svn co http://svn.servidor.net/proyecto/trunk* -->Recepción por primera vez.
- *svn checkout http://svn.servidor.net/proyecto/trunk <ruta>/proyecto* --> Recepción ordinaria.
- *svn status* --> Verificación del estado actual del repositorio local.
- svn log -v -r<version> --> Revisión de cambio entre versiones.
- *svn info index.php* --> Solicitud de información de un archivo específico.
- *svn add <ruta>/class.php* --> Agregar un archivo al repositorio.
- svn update --> Actualizar las fuentes locales.
- *svn commit -m 'Descripción de la actualización'* --> Envío de modificaciones locales al repositorio.

Existen varias interfaces o programas individuales que sirven para integrar subversión en entornos de desarrollo, algunos ejemplos serian:

- TortoiseSVN. Provee integración con el explorador de Microsoft Windows. Es la interfaz más popular en este sistema operativo.
- Subclipse. Complemento que integra Subversion al entorno Eclipse.
- Subversive. Complemento alternativo para Eclipse.
- ViewVC. Interfaz web, que también trabaja delante de CVS.
- SvnX. Proporciona integración con Mac OS X.
- RapidSVN. Proporciona integración con Linux, y también con Mac OS X.
- RabbitVCS, para el administrador de archivos Nautilus del escritorio GNOME.
- KDESvn. Provee integración con el entorno de escritorio KDE.
- Easyeclipse, es un paquete basado en Eclipse, con algunos complementos de código abierto.
- Versions. Interfaz de escritorio para Mac OS X.
- AnkhSVN. Extensión para Visual Studio.

En nuestro proyecto vamos a utilizar uno de estos clientes, más concretamente subclipse, por lo que lo describiremos con más detalle.

Subclipse existe para proporcionar una interfaz de usuario excepcional a Subversion desde el IDE de Eclipse. Su objetivo es proporcionar un cliente que sea tan robusto y fácil de usar como el cliente de CVS que viene con Eclipse. Dicho esto, hay una gran cantidad de importantes diferencias técnicas entre Subversion y CVS. Por lo tanto, en subclipse se trata de encontrar un equilibrio entre el suministro de una interfaz de usuario que sea familiar para los experimentados usuarios de CVS de Eclipse, y uno que sea apropiado para la acción a la que Subversion está expuesta. En particular, la forma en que Subversion maneja las ramas y etiquetas es muy diferente de la CVS. En

consecuencia, la interfaz de usuario de estas características en Subclipse es diferente de la interfaz de usuario para el cliente de CVS.

1.5. APACHE TOMCAT

Apache Tomcat es un software de código abierto que implementa la tecnología Java Servlet y JavaServer Pages. Las especificaciones del Java Servlet y JavaServer Pages se han desarrollado bajo la Comunidad de Procesos de Java.
Tomcat es un servidor web con soporte de Servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en Servlets. El motor de Servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.
Tomcat puede funcionar como servidor web por sí mismo. Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.
Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.
Apache Tomcat es desarrollado en un entorno abierto y participativo, publicado bajo la licencia Apache versión 2. Apache Tomcat está destinada a ser una colaboración de los desarrolladores de la mejor clase de todo el mundo.
Apache Tomcat controla numerosos aplicaciones web de misión crítica a través una amplia gama de industrias y organizaciones.
Las primeras distribuciones de Tomcat fueron las versiones 3.0.x hasta la más recientes la 7.x. A partir de la versión 4.0 Tomcat utiliza el contenedor de Servlets Catalin. Las características de Tomcat según su versión son:

Tomcat 3.x (distribución inicial)

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de Servlets
- Funciones básicas HTTP

Tomcat 4.x

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de Servlets rediseñado como Catalina
- Motor JSP rediseñado con Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

Tomcat 5.x

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Análisis rápido JSP

Tomcat 6.x

- Implementado de Servlet 2.5 y JSP 2.1
- Soporte para *Unified Expression Language* 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores
- Soporte para Comet a través de la interfaz CometProcessor

Tomcat 7.x (Versión actual)

- Implementado de Servlet 3.0 JSP 2.2 y EL 2.2

- Mejoras para detectar y prevenir "fugas de memoria" en las aplicaciones web
- Limpieza interna de código
- Soporte para la inclusión de contenidos externos directamente en una aplicación web

Me voy a centrar en la versión de **Tomcat 5.5** porque es la que he utilizado para el desarrollo del proyecto.
Las versiones de Apache Tomcat 5.5 implementa el Servlet 2.4 y JavaServer Pages 2.0 especificaciones del Java Community Process , e incluye muchas características adicionales que hacen que esta plataforma sea útil para el desarrollo e implementación de aplicaciones y servicios web.
En la documentación de Tomcat hay muchas referencia a variables $CATALINA_HOME. Esta representa la raíz de tu instalación de Tomcat. En mi caso $CATALINA_HOME es igual a "/home/pfc/Archivos/tomcat". Cuando decimos: "El $CATALINA_HOME/README.txt", esto quiere decir que el archivo README.txt está en la raíz de instalación de Tomcat, es decir, el archivo "/home/pfc/Archivos/tomcat/README.txt".
Estos son algunos de los directorios clave de Tomcat, todos dentro de $CATALINA_HOME:

- **/ bin:** donde se encuentra los archivos compilados de Tomcat para su arranque, parada, y otros scripts que necesita. Según el sistema operativo en el que se ejecuta el Tomcat hay dos tipos de archivos: los archivos *. sh para los sistemas Unix y archivos *. bat para sistemas Windows. Para Windows como consola de comandos de Win32 carece de algunas funciones, hay algunos archivos adicionales de aquí.
- **/ conf:** Aquí se localizan los archivos de configuración y DTD relacionados. El archivo más importante aquí es server.xml. Este es el archivo de configuración principal para el Apache Tomcat.
- **/ logs:** Aquí se sitúan los archivos de registro de todo lo que se hace y sucede en tomcat.
- **/ webapps:** Aquí es donde se localizan todas las aplicaciones web que se ejecutan en Tomcat.

Instalación de Apache Tomcat:

Windows

La instalación de Tomcat en Windows se puede hacer fácilmente utilizando el instalador de Windows. Su interfaz y el funcionamiento es similar a los instaladores basada en wizard.
Tomcat se instala como un servicio de Windows, sin importar la configuración seleccionada. Puedes configura el servicio como inicio automático cuando se inicia Windows marcando el checkbox "auto" durante la instalación. Para una seguridad óptima, el servicio se debe ejecutar como un usuario independiente, con permisos reducidos.
Tomcat necesita un JRE para ejecutar el servicio. El programa de instalación utiliza el registro para determinar la ruta de la base de un JRE. Cuando se ejecuta en un sistema operativo de 64 bits, el instalador buscará primero un JRE de 64 bits y sólo buscan un JRE de 32 bits si el JRE de 64 bits no se encuentra. No es obligatorio utilizar el JRE por defecto detectado por el instalador. Usted puede seleccionar cualquier JRE instalado como Java 5 o posterior 32-bit o 64-bit. Cuando Tomcat se ejecuta como un servicio, habrá un icono de la bandeja visible.

Linux

La instalación de Tomcat en Linux es muy simple solo hay que descomprimir un archivo apache-tomcat-5.5.35.tar.gz en el directorio "/opt". Para iniciar el Apache Tomcat se hace ejecutando el archivo $CATALINA_HOME/bin/startup. Para pararlo se hace ejecutando el archivo $CATALINA_HOME/bin/shutdown. Para iniciar el servicio automáticamente necesitamos la herramienta jsvc del proyecto commons-daemon. Antes de ejecutar el script debe asegurarse que la variable de entorno JAVA_HOME apunta a la base del JDK. Con los comando que veremos a continuación debería dar como resultado un archivo binario jsvc, ubicado en el directorio $CATALINA_HOME/bin. Esto supone que utilizar GNU TAR, y que CATALINA_HOME es una variable de entorno que apunta a la ruta de la base de la instalación de Tomcat.

```
cd $CATALINA_HOME/bin
tar xvfz commons-daemon-native.tar.gz
cd commons-daemon-1.0.x-native-src/unix
./configure
make
cp jsvc ../..
cd ../..
```

El archivo $ CATALINA_HOME/bin/commons-daemon-1.0.x-native-src/unix/native/Tomcat5.sh se puede utilizar como una plantilla para iniciar Tomcat automáticamente en el arranque de /etc/init.d.

1.6. SERVLET

Los Servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGIs (common gateway interface), programas que generan páginas web en el servidor, o los ASP (Active Server Pages), un método específico de Microsoft. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y Servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada Servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Los Servlets son objetos que se ejecutan dentro del contexto de un contenedor de Servlets por ejemplo Tomcat y extienden su funcionalidad.

La palabra Servlet proviene de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador Web.

Un Servlet es un objeto que se ejecuta en un servidor o contenedor JEE, diseñado especialmente para brindar contenido dinámico desde un servidor Web, generalmente HTML. Forman parte de JEE (Java Enterprise Edition), que es una ampliación de JSE (Java Standard Edition).

Un Servlet implementa la interfaz javax.servlet.Servlet o hereda alguna de las clases más convenientes para un protocolo en específico. Al implementar esta interfaz el Servlet es capaz de interpretar los objetos de tipo HttpServletRequest y HttpServletResponse quienes contienen la información de la página que invocó al Servlet.

Ciclo de vida de un Servet

El ciclo de vida de un Servlet se divide en los siguientes pasos:

- El cliente solicita una petición a un servidor vía URL.
- El servidor recibe la petición.
    - Si es la primera, se utiliza el motor de Servlets para cargarlo y se llama al método init().
    - Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
- Se llama al método service() para procesar la petición devolviendo el resultado al cliente.
- Cuando se apaga el motor de un Servlet se llama al método destroy(), que lo destruye y libera los recursos abiertos.

Clases y objetos necesarios

Se puede crear un Servlet haciendo uso del paquete java.servlet que brinda las clases y objetos necesarios para el trabajo con Servlets.

Ventajas de los Servlets versus CGI.

- Los Servlets Java son más eficientes, fáciles de usar, más potentes, más portables y más baratos que el CGI tradicional y otras muchas tecnologías del tipo CGI.
- Eficiencia. Con CGI tradicional, se arranca un nuevo proceso para cada solicitud HTTP. Si el programa CGI hace una operación relativamente rápida, la sobrecarga del proceso de arrancada puede dominar el tiempo de ejecución. Con los Servlets, la máquina Virtual Java permanece activa, y cada petición es manejada por un hilo Java de peso ligero, no un pesado proceso del sistema operativo. En CGI tradicional, si hay n peticiones simultáneas para el mismo programa CGI, el código de este problema se cargará n veces en memoria. Sin embargo, con los Servlets, hay n hilos pero sólo una instancia de la clase Servlet. Los Servlets también tienen más alternativas que los programas normales CGI para optimizaciones como los cachés de cálculos previos, mantener abiertas las conexiones de bases de datos, etc.
- Conveniencia. Los Servlets tienen una gran infraestructura para el análisis automático y decodificación de datos de formularios HTML, leer y seleccionar cabeceras HTTP, manejar cookies, seguimiento de sesiones, y muchas otras utilidades.
- Potencia. Los Servlets Java nos permiten hacer muchas cosas que son difíciles o imposibles con CGI normal. Por algo, los Servlets pueden intercambiar directamente con el servidor Web. Esto simplifica las operaciones que se necesitan para buscar imágenes y otros datos almacenados en situaciones estándares. Los Servlets también pueden compartir los datos entre ellos, haciendo las operaciones útiles como almacenes de conexiones a bases de datos fáciles de implementar. También pueden mantener información de solicitud en solicitud, simplificando operaciones como seguimiento de sesión y el caché de cálculos anteriores.
- Portable. Los Servlets están escritos en Java y siguen un API bien estandarizado. Consecuentemente, los Servlets escritos, digamos en el servidor I-Planet Enterprise, se pueden ejecutar sin modificarse en Apache, Microsoft IIS, o WebStar. Los Servlets están soportados directamente o mediante plug-in en la mayoría de los servidores Web.
- Barato. Hay un número de servidores Web gratuitos o muy baratos que son buenos para el uso personal o el uso en sitios Web de bajo nivel. Sin embargo, con la excepción de Apache, que es gratuito, la mayoría de los servidores Web comerciales son relativamente caros. Una vez que tengamos un servidor Web, no importa el coste del servidor, añadirle soporte para Servlets (si no viene pre configurado para soportarlos) es gratuito o muy barato.

Ejemplo código de ejemplo de un Servlet que procesa una petición GET y devuelve una página web HTML sencilla:

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HolaMundoServlet extends HttpServlet {

   @Override
   public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      PrintWriter out = response.getWriter();
      out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">");
      out.println("<html>");
      out.println("<head><title>Ejemplo Hola Mundo</title></head>");
      out.println("<body>");
      out.println("<h1>¡Hola Mundo!</h1>");
      out.println("</body></html>");
   }
}
```

Listado 1. Código ejemplo Servlet

1.7. Java Server Pages

Java Server Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI,

incluyendo los Servlets, hacen que generemos la página completa mediante nuestro programa, incluso aunque la mayoría de ella sea siempre lo mismo.

Para empezar, los JSPs y Servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada Servlet o JSP, a partir de ahora lo usaremos de forma indistinta se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia le permite también hacer una serie de cosas de forma más eficiente como la conexión a bases de datos y el manejo de sesiones.

Los JSPs son en realidad Servlets, esto quiere decir que un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un Servlet. La principal diferencia entre los Servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un Servlet es un programa que recibe peticiones y genera a partir de ellas una página web

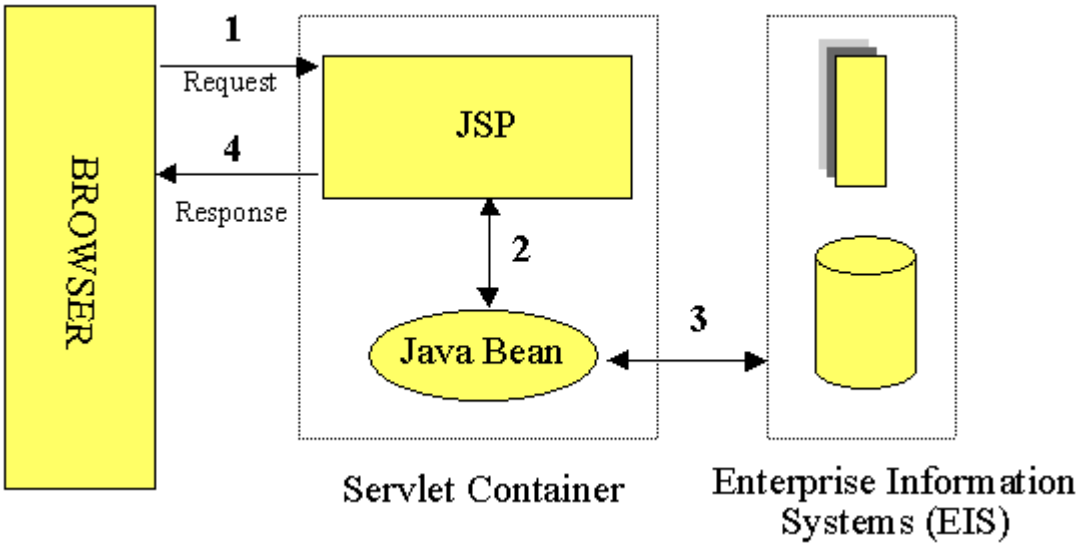

Figura 1: Esquema de JSP

Ventajas de los JSP

- Contra Active Server Pages (ASP). ASP es una tecnología similar de Microsoft. Las ventajas de JSP están duplicadas. Primero, la parte dinámica está escrita en Java, no en Visual Basic, otro lenguaje específico de MS, por eso es mucho más poderosa y fácil de usar. Segundo, es portable a otros sistemas operativos y servidores Web.

- Contra los Servlets. JSP no nos da nada que no pudiéramos en principio hacer con un Servlet. Pero es mucho más conveniente escribir (y modificar!) HTML normal que tener que hacer un billón de sentencias println que generen HTML. Además, separando el formato del contenido podemos poner diferentes personas en diferentes tareas: nuestros expertos en diseño de páginas Web pueden construir el HTML, dejando espacio para que nuestros programadores de Servlets inserten el contenido dinámico.

- Contra Server-Side Includes (SSI). SSI es una tecnología ampliamente soportada que incluye piezas definidas externamente dentro de una página Web estática. JSP es mejor porque nos permite usar Servlets en vez de un programa separado para generar las partes dinámicas. Además, SSI, realmente está diseñado para inclusiones sencillas, no para programas "reales" que usen formularios de datos, hagan conexiones a bases de datos, etc.

- Contra JavaScript. JavaScript puede generar HTML dinámicamente en el cliente. Este una capacidad útil, pero sólo maneja situaciones donde la información dinámica está basada en el entorno del cliente. Con la excepción de las cookies, el HTTP y él envió de formularios no están disponibles con JavaScript. Y, como se ejecuta en el cliente, JavaScript no puede acceder a los recursos en el lado del servidor, como bases de datos, catálogos, información de precios, etc.

JSP nos permite crear dos partes de forma separada. Aquí tenemos un ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Welcome to Our Store</TITLE>
</HEAD>
<BODY>
      <H1>Welcome to Our Store</H1>
      <SMALL>Welcome, <!-- User name is "New User" for first-time visitors -->
            <% out.println(Utils.getUserNameFromCookie(request)); %> To access
            your account settings, click <A HREF="Account-
Settings.html">here.</A></SMALL>
      <P>Regular HTML for all the rest of the on-line store's Web page.
</BODY>
</HTML>
```

Listado 2. Ejemplo de código de una página JSP

CAPITULO 2. PROYECTO SAKAI

Proyecto Sakai es una comunidad que está creando una tecnología que mejora la enseñanza, el aprendizaje y la investigación.

2.1. La Comunidad

La comunidad global se reúne para definir las necesidades de los usuarios académicos, crear herramientas de software, compartir las mejores prácticas y compartir conocimientos y recursos en soporte de este objetivo. Cada día miembros de la comunidad comparten miles de interacciones - la construcción y mejora del software, para solicitar ayuda, colaborando en proyectos, y disfrutar de las relaciones que se derivan de este trabajo. Si bien reconocemos que a veces es difícil encontrar el punto correcto para entrar en la comunidad. Cada nuevo participante aporta conocimiento y una perspectiva que nos beneficiará a todos.

Organizaciones que usan Sakai

La comunidad se compone de más de 350 organizaciones de diversos perfiles en todo el mundo. Estas están adoptando, contribuyendo, compartiendo y apoyando el proyecto Sakai. Algunas de estas organizaciones son la universidad Politécnica de Valencia, la universidad de Oxford, ...

Contribución

La comunidad busca reconocer las contribuciones de voluntarios que participan en la mejora del software y la comunidad. Mientras que aquí se destacan unas pocas categorías, existen muchos particulares trabajando para fortalecer Sakai a lo largo del mundo. Sakai se administra por:

- Los compañeros de Sakai comparten las contribuciones pendientes.
- El comité de coordinación técnica CLE asegura una publicación a tiempo y de calidad.
- El equipo de mantenimiento CLE resuelve los problemas tras la publicación.
- El grupo directivo de Sakai OAE supervisa la gestión del desarrollo del proyecto.

1. Comité de Coordinación Técnica de Sakai CLE (TCC).
   Su objetivo del TCC es crear una publicación a la comunidad fiable y de alta calidad en la fecha prevista. La TCC coordina con grupos y procesos de la comunidad, de tal forma proporcionar una dirección técnica, asesoramiento y coordinación para Sakai CLE.
   El TCC es un grupo abierto, todas las comunicaciones son públicas y cualquier persona es bienvenida a contribuir. El equipo está compuesto principalmente por los miembros experimentados de la comunidad los cuáles aportan varias perspectivas y experiencias para orientar y apoyar a los procesos de la comunidad. Algunos miembros son: Alan Berg, Noah Botimer, Matthew Buckett, …

2. Miembros Sakai
   El programa de Miembros Sakai fomenta el liderazgo de la comunidad mediante el reconocimiento de los miembros voluntarios más destacados. Los Miembros Sakai aportan una enorme experiencia a la comunidad en las áreas de diseño y desarrollo de software,

prácticas pedagógicas y de enseñanza y apoyo a la comunidad. Desde 2008, seis Miembros Sakai son seleccionados anualmente por un pequeño comité compuesto por miembros de la comunidad Sakai para realizar un curso de un año de duración. Los Miembros reciben una modesta remuneración económica para ayudar a costear las actividades relacionadas con Sakai, como por ejemplo: viajes, conferencias, equipamiento y gastos personales. Actualmente son : Rob Coyle de la universidad de Johns Hopkins, Carl Hall de Hallway Technologies, David Roldán Martínez de la Universidad Politécnica Valencia, Megan May y Brian Richwine de la universidad de Indiana, Zhen Qian, Universidad de Michigan.

3. Equipo de mantenimiento de Sakai CLE

   El equipo de mantenimiento soluciona errores y mejora la calidad de la versión operativa de Sakai CLE. El equipo de mantenimiento es un grupo de miembros de la comunidad que trabajan para resolver todos los errores del producto Sakai CLE. Este grupo vela por la arquitectura del producto, soluciona errores, aplica parches y ayuda en la gestión del sistema de seguimiento de errores. Suelen ser los primeros en responder a las preguntas de los recién llegados en la lista de correo y asisten en la solución de problemas. Es un grupo amigable que siempre está buscando nuevos miembros para ocuparse de tareas que crean un gran impacto positivo en la comunidad. Algunos Miembros del equipo de mantenimiento son: Matthew Buckett, Jonathan Cook, Nuno Fernandes, David Horwitz,…

4. Grupo directivo Sakai OAE

   El grupo directivo supervisa la gestión del desarrollo del Entorno Académico Abierto (Open Academic Environment OAE). El grupo directivo de Sakai OAE proporciona supervisión al proyecto teniendo en cuenta las necesidades de las instituciones colaboradoras y sus requisitos para ofrecer a la comunidad publicaciones con éxito.

   Dado el enfoque del proyecto, el grupo directivo cumple un rol importante en el soporte de la eficacia del proyecto OAE, el director de Alan Marks.

   Cada organización representada en el grupo directivo ha realizado una inversión de recursos en el proyecto OAE significativa. No se requiere la participación en el equipo de directivo por parte de las organizaciones investigadoras.

   Los miembros del grupo directivo Sakai OAE son:

   a. David Goodrum – Universidad de Indiana, Estados Unidos (Presidente)

      A día de hoy la universidad de Indiana utiliza con éxito CLE para dar servicio a más de 100.000 estudiantes a lo largo de ocho campus. El compromiso de la UI en el proyecto Sakai OAE se centra especialmente en traer los puntos fuertes de diseño de instrucción de la universidad al proyecto y alisar el camino a las instituciones que migran de Sakai CLE o usan ambos Sakai CLE y AOE.

   b. Philip Uys - Universidad Charles Sturt, Australia. La universidad Charles Sturt es una universidad internacional cuya misión se centra en la difusión flexible de la enseñanza. El conjunto de emplazamientos, métodos de difusión y programas de

estudio soportados por la universidad hacen que tenga necesidades únicas en cuanto a flexibilidad del sistema de enseñanza y colaboración.

c. David Ackerman - Universidad de New York , Estados Unidos. La universidad de New York abarca una gran cantidad de casos de uso que no cumple cualquier LMS existente. Las necesidades están conducida por varias iniciativas escolares para crear una comunidad académica más fuerte e incrementar la acción y el compromiso de los estudiantes en su experiencia académica.

d. Clay Fenlason - Instituto tecnológico de Georgia, Estados Unidos. La capacidad técnica del instituto tecnológico de Georgia tuvo como resultado una adopción limitada de su sistema tradicional de gestión de la enseñanza. El instituto busca un entorno tecnológico flexible que permita a los innovadores desarrollar y combinar herramientas y contenido y compartir su trabajo de forma sencilla a través de la universidad y la comunidad.

   Oliver Heyer – Universidad de Berkeley en California, Estados Unidos. Del mismo modo que la universidad de Indiana, Berkeley usa Sakai CLE eficientemente para dar soporte a la enseñanza, el aprendizaje, la investigación y la colaboración para sus estudiantes. Su necesidad inmediata es crear un portal de estudiantes que reúna todos los recursos de la institución que asisten al éxito estudiantil individual. Berkeley usará las capacidades de Sakai OAE para la creación de contenidos de autor y recombinaciones flexibles para conseguirlo.

e. John Norman – Universidad de Cambridge, Reino Unido. El entorno académico único que requiere la universidad de Cambridge es un sistema que permita a un estudiante explorar, sintetizar y crear contenido académico. Esta experiencia de aprendizaje se da en un entorno rico en recursos, expertos y artefactos pero a la luz de límites y procesos definidos. La participación de Cambridge está formada por personal del Centro de investigación aplicada, una organización que busca definir nuevos sistemas y modelos para dar soporte a esta experiencia de aprendizaje e investigación flexibles.

   Sean DeMonner - Universidad de Michigan, Estados Unidos. Como parte del esfuerzo de NextGen Michigan, la universidad está buscando como añadir Sakai OAE a su ecosistema de servicios de tecnología académica. Hay varios objetivos de alto nivel en el proyecto los cuáles son: equilibrio entre seguridad y apertura, integrarlo de forma libre y modular, soportar la distinción y el control claros sobre la dualidad de datos privados y públicos, ser de confianza a escala, pero flexible y sensible a las necesidades cambiantes del negocio y facilitar el descubrimiento de contenidos mientras se refuerza la política de propiedad intelectual. El entorno de

tecnología académica resultante será vital para dar soporte a la enseñanza, el aprendizaje, la investigación y servicios de la Universidad.

Compartición entre la Comunidad

Las alineaciones creadas alrededor del software de Sakai permite a los miembros de la comunidad compartir una variedad de prácticas y herramientas. Organizaciones a lo largo de la comunidad comparten prácticas, procesos, herramientas y tecnología. El grupo de Enseñanza y Aprendizaje de Sakai comparte prácticas, métodos y experiencias a lo largo de la comunidad. Las instituciones comparten información de modo simple y amplio para fomentar los procesos de innovación y mejorar el recorrido de cada uno.

1. Comunidad de enseñanza y aprendizaje Sakai
   La comunidad T&L (Teach and Learn) mantiene una reunión virtual de una hora cada mes para coordinar iniciativas y una reunión virtual trimestral para actualizar y solicitar retroalimentación de la comunidad. Fuera de estas reuniones hay pequeños grupos que colaboran en iniciativas clave.
   El grupo comparte su trabajo en la web de "OpenEd Practices" la cual incluye grupos de discusión, ejemplos de trabajo e información de proyecto.

   Se celebran anualmente unos premios a la innovación en la enseñanza. Estos animan a compartir y reconocer prácticas y pedagogía excepcionales. El premio a la innovación en la enseñanza con Sakai (TWSIA) es sólo una de las contribuciones de la Comunidad de enseñanza y aprendizaje. Pese a que existen muchos aspectos en los cuales la tecnología y en particular Sakai, pueden hacer el proceso de enseñanza más productivo o eficiente, las aplicaciones tecnológicas innovadoras realmente transforman la experiencia educacional. La intención de este premio es destacar ejemplos de aplicaciones de Sakai a la educación que entran en la categoría de innovadoras o revolucionarias. En los premios de 2010 se presentó el uso de Sakai para la enseñanza y el aprendizaje en la conferencia de Sakai en Denver, Colorado del 15 al 17 de Junio de 2010. Los ganadores fueron:

   - Primer premio fue para: Scott Bowman
   - Segundo premio fue para: Sally Knipe
   - Las Menciones Honorables fueron para: Joshua Danish y Karen Swenson

   Otro grupo destacado dentro de la comunidad de T&L es el grupo de trabajo en diseño de objetivos de aprendizaje de Sakai.

Los objetivos de diseño se usan para informar el nuevo desarrollo de Sakai CLE y Sakai OAE. Los objetivos y facetas apuntan a capturar no sólo lo que profesores y alumnos necesitan o quieren hoy en términos de capacidad de aprendizaje sino lo que como comunidad creemos que van a necesitar y querer en el futuro. En el sentido de que estos promuevan no sólo un uso tradicional de CLE sino otros usos más innovadores de cara al futuro. La capacidad de soportar flujos de trabajo de portfolio es un factor clave de especial atención para el nuevo Entorno Académico Abierto de Sakai (OAE). Los grupos de enseñanza y aprendizaje enfatizan la necesidad de capacidades integradas con la creación de portfolios que incluyan recopilación, reutilización, etiquetado, organización y reflexión sobre artefactos de aprendizaje a través de todo el entorno. Imagen, escenarios de usuario, manifiestos y recorridos de usuario son posibilidades de trabajo para el futuro. Los grupos de enseñanza y aprendizaje querrían centrarse en aquello que los diseñadores, desarrolladores, consejo de producto y la comunidad en general considera más útil. Cada objetivo de diseño no es un área de funcionalidad sino una perspectiva desde la cual todo el sistema debe ser considerado. Las facetas se definen en genera y también en relación a los resultados deseados en Sakai. Todos los objetivos y facetas han de ser tenidos en cuenta cuando se trabaja en nuevas funcionalidades de Sakai.

2. Caso de estudio de “Brook University “

Brock University es una universidad de más 18.000 estudiantes que se está convirtiendo en su estado de nuevo y exhaustivo. Se encuentra ubicado en la región del Niágara de Canadá y su nombre se lo debe a Sir Isaac Brock, un general valeroso importante que jugó un papel decisivo en la defensa de Alto Canadá contra los Estados Unidos durante la guerra de 1812.

Piloto y transición

En 2006 Rector Universidad de Brock, y el CIO de la Universidad de cargo del Centro de Enseñanza, Aprendizaje y Tecnologías Educativas (CTLET) para evaluar el futuro de WebCT CE 6 y otras alternativas de código abierto, como Sakai y Moodle para el uso como principal sistema de gestión del aprendizaje de Brock (LMS) a partir del año académico 2009. Como WebCT Inc. ha sido recientemente adquirida por “Blackboard University” quería hacer una elección entre quedarse con las actuales basadas en proveedor LMS o migrar a un LMS abiertos que ofrecen mayor portabilidad y la interoperabilidad de la enseñanza y el aprendizaje de contenidos. El grupo asesor opto por coordinar un programa piloto, sintiendo que ambos tenían una oferta mucho más ventajoso y, al mismo tiempo era relativamente desconocido. Se puso también a prueba Moodle y se pusieron en contacto con otras universidades canadienses acerca de su experiencia con Moodle como uno de sus LMS. El piloto de Sakai 2.3 se realizó durante el año académico 2007. Los profesores tenían la opción de incluir el curso que se enseña en este programa piloto y 50 cursos y 27 instructores optaron por tomar parte. Todos los estudiantes de la Brock University fueron capaces de entrar en el sistema piloto de Sakai. Un estudio formal

de los instructores y estudiantes participantes se llevó a cabo. Los estudiantes y profesores apreciaron la simplicidad de Sakai, con instructores tomando nota de la baja barrera de entrada y se vio que el nivel de complejidad de Sakai era proporcional a la complejidad del curso en línea. Los instructores también vieron Sakai como se adaptaba mucho mejor a cursos híbridos, que era el principal motivo porque la "Brock University" utilizaba los LMS. El grupo que llevó a cabo el piloto presentó los resultados al Senado y al Rector de la Universidad. El Rector y Vice-Presidente decidieron implementar un sistema basado en Sakai como sistema primario basado en LMS a partir del año académico 2008, pasando a ser el único LMS usado en esta universidad en el curso 2009.

Implementación

En el otoño de 2008 y el invierno de 2009 "Brock University" tenía dos LMS que se ejecutan en la producción: WebCT y Sakai. Fue en su último año de la licencia para ejecutar WebCT, Blackboard Learning System para la "Brock University", ya que esta se encontraba en el proceso de transición a Sakai. Una importante inversión se hizo en el hardware para ejecutar el nuevo sistema de Sakai, que refleja el compromiso de la universidad a la mejora de los LMS más de aumento de las necesidades para el sistema de Sakai. Estos sistemas paralelos eran un inconveniente menor para la coordinación del personal a los estudiantes y profesores, ya que hizo lo hizo a facilitar la transición, por otra parte el contraste de toda la experiencia de relieve la necesidad del cambio. Sólo era económicamente viable tener dos sistemas paralelos en la producción debido a que el nuevo sistema de código abierto Sakai no se requiere ningún pago de licencias. Los instructores podían elegir si querían seguir utilizando WebCT / Blackboard por un último año o empezar de nuevo en Sakai. Durante el otoño de 2008 y el invierno de 2009 período en que ambos LMS se ejecutaban en paralelo al CTLET trabajó con un pequeño equipo de tiempo de los estudiantes para realizar la conversión de los cursos de WebCT a Sakai 2.5. Todos los cursos de 2007 se convirtieron en sitios de Sakai. Los instructores fueron contactados para que visitasen un sitio web en el que indicaran cuáles de sus cursos anteriores les gustaría que se migrara a Sakai. Los estudiantes, que trabajan a tiempo parcial desde su propia casa, había completado la conversión de un plazo de seis meses. A principios del 2009 año académico de la Universidad de Brock sistema de Sakai, conocido localmente como Isaak se convirtió en único LMS de Brock.

Uso actual

En el año académico 2010-2011 la Brock University tenía 2825 sitios de los cursos creados en Sakai. Esto equivale a aproximadamente tres cuartas partes de todos los cursos de crédito total y parcial en la universidad.

Reflexiones

Isaak, Sakai basados en la Brock University de LMS ha permitido a la Brock University de integrar pedagogías innovadoras en la enseñanza y en el aprendizaje, tanto en los

elementos que se entregan en línea y aquellos que se entregan fuera de línea. La “Brock University” se ha beneficiado de un LMS, que ofrece muchas características que apoyan la enseñanza y el aprendizaje de todo listo para ir tan pronto como el instructor les necesita. La “Brock University” se ha tratado con algunos bugs menores con versiones publicadas de Sakai, sin embargo las respuestas rápidas de la comunidad en general y la tranquilidad de no tener nada que no pudiera ser investigado localmente en el sistema ha sido tranquilizador. Además sabiendo que la institución tiene la opción, si así lo decide, puede permanecer en cualquier versión idiosincrásica de Sakai que pone de forma indefinida los conjuntos de los estudiantes, el personal, los profesores y las administradores en el control de nuestra propia relación con el LMS.

El sistema robusto y fiable se ha asegurado que los estudiantes y los profesores están dispuestos a incluirlo en las partes críticas de los cursos, tales como el trabajo en grupo y la evaluación. La variedad y flexibilidad de la herramienta Sakai viene con, y la posibilidad de añadir herramientas de las contribuciones de otras universidades, como la herramienta de registro. Al mismo tiempo, la falta de suposiciones sobre la estructura de un curso y en algunos casos un conjunto de características mínimas, fomenta la integración de otras herramientas cuando sea apropiado y en su mayor parte, esto es fácil de hacer. Después de la transición el aumento de los recursos por parte del departamento de TI y la CTLET es proporcional el crecimiento en el uso, e involucra a los mismos individuos apoyo en el sistema comercial anterior. El diseño sencillo y directo que se comporta como todos los sitios web modernos lo hacen otros, y permite que el contenido de enseñanza y actividades de aprendizaje a ser el foco. En realidad, puede ser sólo superficial, pero la simplicidad de Isaak / Sakai, que fue comentado durante el proceso piloto, sigue siendo válida hoy en día.

Soporte

Las Instituciones emplean un amplio rango de métodos y recursos para mantener Sakai. Dado que Sakai se distribuye como Recurso de Software Libre y Abierto, las organizaciones de Sakai tienen muchas opciones de soporte. Algunos eligen instalar, alojar y mantener el software por ellos mismos con la ayuda del resto de la comunidad Sakai, mientras que otros eligen trabajar con suministradores comerciales que ofrecen los servicios de Sakai.

1. Soporte de la comunidad

Hay una red en activo de instituciones de la comunidad que comparten el soporte de Sakai. Los miembros de la comunidad Sakai comparten su experiencia y proporcionan soporte mutuo diariamente. Hay principalmente tres foros:

- El primero es un conjunto de listas de correo gestionadas por la Fundación Sakai dando soporte a las actividades de colaboración de la comunidad. Para la asistencia a preguntas técnicas el grupo más activo es el de discusión de Desarrollo (Sakai-Dev), pero hay otros muchos disponibles desde enseñanza y aprendizaje hasta experiencia del usuario. Visite la página de Listas de correo para más información.
- El segundo es una wiki usada por los miembros de la comunidad y gestionada por la fundación para una gran variedad de fines incluyendo colaboración con la documentación o ideas de diseño, gestión de proyectos y la difusión de notas a través de los temas. También se dispone de documentación técnica online para Sakai.
- El tercero es un sistema de seguimiento de tareas gestionado por la fundación y usado por los miembros de la comunidad para registrar peticiones de funcionalidad, generar nuevos requisitos y localizar y resolver errores.

2. Servicio Comercial

Los Afiliados de Sakai Comercial proporcionan un valioso soporte y una importante contribución a la comunidad también proporcionan un apoyo comercial para las instituciones que usan Sakai. Contribuyen con una financiación valiosa a la Fundación y directamente apoya al trabajo de desarrollo y de programas especiales. La SCA proporciona; servicios y apoyo, la integración a los servicios y a los contenidos, y al desarrollo personalizado. Algunos Socios son: Asteros, Edia, Embanet Compass Academic Services, IBM, KEL.

2.2. Sakai como Producto

Sakai es un software adecuado desarrollado por la comunidad. Históricamente, la comunidad se hizo cargo de un único proyecto, el entorno de aprendizaje y colaboración de Sakai(CLE). Este proyecto se presentó también en el portafolio (OSP), incluido en Sakai (CLE). Hoy, mientras continúa mejorando el entorno de aprendizaje de Sakai (CLE) la comunidad está desarrollando un nuevo proyecto, el entorno académico Sakai (OAE) sobre el enfoque de colaboración académica.

Sakai CLE.

Es un sistema para la enseñanza, el aprendizaje, la investigación y la colaboración mediante la tecnología con funcionalidad total.

1. CLE para la gestión del aprendizaje
   Sakai CLE proporciona un entorno para la enseñanza y el aprendizaje basado en la colaboración y el conocimiento compartido. La comunidad Sakai ha identificado un conjunto de capacidades "básico" adoptado por prácticamente todas las instituciones que usan Sakai. La comunidad prueba, gestiona y mantiene estas capacidades a través del ciclo

de publicación de la comunidad. Las herramientas básicas se localizan en la versión 2.7 y son Anuncios, Asignaciones, Blog, Calendario, Chat, Foro, Dropbox, Archivos de email, Glosarios, Calificaciones, Noticias, Perfiles, Recursos, Estadísticas de los sitios, Plan de estudios, Exámenes, Cuestionarios, Página web y una wiki. Además, hay un gran número de herramientas adicionales llamadas "contrib" disponibles que han sido desarrolladas por la comunidad y son usadas por muchas instituciones.

2. CLE para colaboración en investigación.
   Los investigadores usan Sakai para colaborar y compartir información y herramientas.
   Sakai CLE ha sido muy usado por personal facultativo en instituciones de investigación para colaborar con otros investigadores. Su uso está en aumento en redes de investigación y organizaciones de enseñanza superior para compartir información y comunicarse a través de la geografía, las organizaciones y las disciplinas. Sakai reduce la carga administrativa sobre los facultativos uniendo el trabajo de investigación y enseñanza. Las característica que soporta CLE para colaboración en investigación son: Anuncios, Blog, Calendario, Chat, Foro, Email, Glosarios, Noticias, Recursos, y Páginas webs.

3. CLE para la colaboración en proyectos.
   Los sitios de proyecto permiten a las organizaciones dar soporte a la colaboración más allá de las aula, facilitando la compartición y la administración sencilla. La capacidad de Sakai CLE para la colaboración es lo suficientemente flexible para dar soporte a una serie de usos. Los espacios para la colaboración en proyectos son usados por grupos de estudiantes, comités facultativos, comités de tesis y líderes de las instituciones implicados esfuerzos de planificación estratégica. Las funcionalidades que dan soporte a estas actividades son Anuncios, Calendario, Chat, Foros, Email, Glosarios, Noticias, Recursos, Páginas Webs, Wiki.

4. CLE para ePortfolios
   Una aplicación robusta de portfolio electrónico con flexibilidad para dar soporte a toda una serie de usos académicos. Los portfolios tienen una amplia gama de usos. Pueden mejorar el proceso de aprendizaje a través de la síntesis y la reflexión, proporcionar un mostrador para logros o dar soporte a la evaluación. El portfolio de Sakai tiene la flexibilidad para satisfacer esta diversidad de necesidades.

   Los estudiantes pueden exhibir su trabajo
   - Recoger los elementos que mejor representan los logros y el aprendizaje
   - Reflexionar sobre estos elementos y su relación
   - Designar un portfolio que muestra la mejor selección de trabajo
   - Publicar el portfolio para un público designado

   Los profesores pueden proporcionar orientación
   - La flexibilidad para los profesores para comprometerse eficazmente con los alumnos
   - Proporcionar una estructura para el trabajo de los alumnos

- Revisar portfolios publicados
- Proporcionar una evaluación formar o una información informal
- Analizar los elementos del portfolio "en conjunto"

Las organizaciones pueden evaluar el aprendizaje de los alumnos

- Las herramientas necesarias para programar la evaluación y la acreditación
- Comparar el rendimiento con los resultados del aprendizaje
- Revisar y analizar el conjunto de resultados del aprendizaje
- Capturar elementos que demuestran el progreso

Sakai OAE (Open Academic Enviroment)

Un completo nuevo sistema que incorpora todos los valores de Sakai CLE, está siendo desarrollado actualmente, y reimagina una nueva visión para la colaboración académica.

Con el propósito de hacer de Sakai el más poderoso e innovador Entorno de Colaboración y Aprendizaje, la comunidad Sakai, con el apoyo y el empuje de la Fundación Sakai, se asegurará de que todas las características del OAE (próxima generación de Sakai) sean accesibles y útiles para el mayor número de usuarios, incluyendo a usuarios con discapacidades.

Para asegurar este alto nivel de accesibilidad para el mayor número de usuarios posible, nuestro objetivo es diseñar OAE para cumplir o exceder los principales diseños de accesibilidad encontrados en las normas de reconocimiento internacional. Nuestro objetivo es cumplir todas las pautas de Accesibilidad de Contenidos de W3C (WCAG) 2.0, niveles A y AA de Criterio de Éxito, y las Pautas de Accesibilidad para la Herramienta de Creación.

Para asegurar que el OAE tiene el menor número de barreras posibles y proporcionar, así, una rica y agradable experiencia a todos los usuarios, Sakai será desarrollado usando normas emergentes y las mejores técnicas de diseño práctico como WAI-ARIA Suite, y tecnologías de adaptación tanto existentes como emergentes.

Expertos en accesibilidad de la comunidad Sakai, seguirán desarrollando las herramientas destinadas para incluirlos en el núcleo CLE, que se someterá a evaluaciones regulares y documentadas de funcionalidad de uso y accesibilidad a través de proceso de diseño y desarrollo para asegurar que se cumplen estos objetivos.

El desarrollo de Sakai OAE modifica la aproximación estándar a la colaboración académica para convertirse:

1. OAE es permeable

   Respetando la relación entre recursos institucionales y las herramientas y contenido externas a la institución. El entorno académico permeable “Si hay algo importante para una educación liberal es el desarrollo de las capacidades de los estudiantes para establecer y llevar a cabo el programa intelectual independientemente en un lugar que se les da.

   Dr. Robert Squillace, Universidad de New York.

   Acceso permeable al mundo del aprendizaje y la investigación permite:

   - Recoger y guardar recursos de dentro y fuera de Sakai
   - Conectar expertos de dentro y fuera de la institución
   - Llevar Sakai a otras plataformas y traer otras plataformas a Sakai
   - Emplear prácticas de enseñanza abierta4

2. OAE es Social

   Facilitando las relaciones que apoyan el aprendizaje y la investigación. En Entorno Académico Social “Lo que conocemos como entorno académico es que las comunidades se unan alrededor del contenido. La capacidad para empezar a mostrar a los estudiantes cómo ocurre eso, les aporta un lugar donde acumular un contenido más amplio, y

comenzar a tener comunidades inteligentes alrededor fue muy importante para nosotros."—Dr. Lucy Appert, New York University.

Conexiones de conocimiento social para el crecimiento académico

- Crear una red de gente dispuesta a responder cuestiones y proporcionar retroalimentación.
- Impulsar la comunicación a través de la red mientras se mantiene completa privacidad y seguridad
- Conectar intereses compartidos con fines educativos
- Fomentar las conexiones entre grupos de estudiantes, profesores e investigadores

3. OAE es Personal

   Otorgar al estudiante propiedad completa y control de la experiencia de aprendizaje.

   Entorno Académico Personal "Antes de que los alumnos sean admitidos, mientras están en la universidad, y potencialmente durante toda su vida, ellos construyen y mantienen ese espacio que los representa como seres intelectuales"—Dr. Robert Squillace, New York University.

   Control personal de su trabajo académico

   - En lugar de mesas de discusión como rígidos contenedores de diálogo, el individuo tiene una habilidad general para comentar, extendiéndose por todo el sistema
   - En lugar de un solo libro de grado, hay una capacidad general para puntuar y graduar por todo el sistema.
   - Crea su ambiente, añadir, borrar, comentar, priorizar contenido y herramientas que pueden o no estar dentro de Sakai.
   - Escoja su punto fuerte y coloque widgets de Sakai en su escritorio web favorito, haciendo que Sakai está disponible dónde usted trabaja
   - Cree sus propias conexiones, cuide su contenido, establezca sus prioridades, cree sus links entre grupos e individuos de confianza.

4. OAE es re mezclable

   Crear una paleta flexible para interactuar con el contenido. El Entorno Académico Re combinable "Considera la facultad de "escribir" un papel. Queríamos entrar en ese proceso, que sabemos que versa sobre pensamiento crítico, análisis y síntesis, el material visual que ellos veían, incorporar eso con el texto, y abrirlo al interior o al exterior para compartirlo más ampliamente." —Dr. Lucy Appert, New York University.

   Contenido recombinadle y conexiones en un lienzo abierto.

   - Crea tus propias experiencias educativas
   - Crea páginas a partir de un lienzo en blanco o a partir de plantillas preparadas.
   - Almacena, re usa, revisa y comparte tu trabajo.
   - Crea aplicaciones web híbridas de funcionalidad, contenido, conexiones y medios

2.3. Fundación

Sakai está apoyado por una fundación que facilita el éxito de la comunidad y su software. La Fundación Sakai no dicta la dirección de la comunidad. Desempeña un papel de apoyo en la creación de la infraestructura de colaboración, lleva el proceso de administración de versiones/publicaciones y facilita la comunicación y la coordinación de toda la comunidad. Cada

miembro de la comunidad Sakai y Sakai software depende de la gestión eficaz de colaboración y de la comunicación y administración que existe en la Fundación. La fundación fomenta la creación de comunidades entre las instituciones, organizaciones con y sin ánimo de lucro y proporciona a sus miembro y otros un marco institucional en el cual pueden florecer proyectos de Sakai. La fundación también trabaja para promover una adopción más amplia de enfoques de estándar abierto y de la comunidad de enseñanza superior.

El personal de la fundación coordina el desarrollo de software, garantiza la calidad y distribuye las actividades en la comunidad. Los miembros del personal supervisan la propiedad intelectual de Sakai y realizan el seguimiento de los acuerdos con contribuidores. Además dan soporte técnico tanto a miembros de la comunidad como a usuarios potenciales hablando en conferencias y otras reuniones sobre Sakai y gestionan las conferencias y eventos propios de Sakai.

La fundación Sakai obtiene soporte a través de contribuidores asociados voluntarios. La organización de miembros de la fundación Sakai elige a los 10 miembros de la junta directiva los cuáles asumen el liderazgo estratégico de la fundación Sakai.

1. Programa de Socios de la fundación Sakai
   El programa de Socios de Sakai provee la base institucional y organizativa para la Comunidad Sakai. Los socios de Sakai son sufragadores de gasto de la fundación Sakai que aportan el capital intelectual, humano y financiero necesario para dar soporte tanto a la fundación como al trabajo de la comunidad. Los socios de Sakai participan en la dirección de la fundación, ayudan a determinar prioridades para la comunidad, y cooperan en cada fase del proceso de producción del software de Sakai. Ser miembro del Programa de Socios de Sakai es opcional y está abierto a instituciones académicas, instituciones sin ánimo de lucro y organizaciones comerciales comprometidas con la visión principal de la Comunidad Sakai de desarrollo y distribución de código fuente abierto.

   Tasas de Afiliación a la Fundación Sakai.

   - Afiliación normal: $10,000 USD por año, renovable anualmente.
   - Afiliación con descuento $5,000 USD por año, renovable anualmente, para instituciones con menos de 3.000 estudiantes de nuevo ingreso cada año.
   - Afiliación Comercial de Sakai (ACS/CSA): las tasas se basarán en una escala. La afiliación costará $2,000 USD por cada $1,000,000 USD en contratos a terceros. La tasa mínima será $2,000 USD y la máxima será $10,000 USD. Para ACSs existentes, esta tarifa se aplicará en la fecha de su renovación como socio comercial.

   Descuento por afiliación de tres años de duración.
   Una afiliación por un periodo de tres años está disponible a un 10% de descuento. Las actuales tasas de afiliación son de $10,000 USD al año para escuelas de gran tamaño, la tasa de afiliación por un periodo de tres años será $27,000 USD. La tasa para escuelas de menor tamaño será de $5,000 USD por año, por lo que por un periodo de tres años se pagaría $13,500 USD. Este descuento por afiliación de 3 años también está disponible para los Afiliados Comerciales de Sakai.

2. Liderazgo

La Junta de la Fundación Sakai y su personal lideran la Fundación. El liderazgo de la comunidad Sakai puede provenir de cualquier individuo. Una sola persona u organización que identifica una oportunidad, puede contribuir con libertad sin necesidad de buscar la aprobación de ningún organismo de dirección. La Fundación Sakai tiene una estructura de liderazgo más definida para asegurar su misión de dar soporte a la comunidad y al software esté garantizada. Todavía, la Junta es elegida por miembros de la Comunidad, específicamente por aquellas instituciones que son miembros de la Fundación Sakai. La Junta, en turnos, supervisa la salud financiera y organizativa. Con esta estructura, la comunidad realmente gobierna la Fundación; la Fundación sirve a la comunidad Sakai.

La Junta de la Fundación es elegida por miembros de la Fundación Sakai. Cada miembro de la Junta ejerce durante un periodo de 3 años y puede ser re elegido para ejercer durante un segundo periodo. La actual Junta está compuesta por líderes de la educación superior tecnológica, respetados miembros de facultades, expertos arquitectos y desarrolladores de software y líderes comerciales.

El personal de la Fundación Sakai, representa un papel muy ligero en el apoyo a la comunidad. El staff no maneja las actividades de la fundación, sino que busca apoyar a los voluntarios haciendo coincidir las necesidades y las capacidades, la conexión de los contribuyentes y facilitando los procesos de comunicación. El personal de la Fundación está siempre disponible para ayudar a los miembros de la comunidad tanto a los nuevos como a los antiguos en los procesos de entendimiento de la comunidad y la participación con los miembros de la comunidad.

CAPITULO 3. INSTALACIÓN SAKAI

Ahora describiremos la instalación de Sakai paso por paso:

Paso 1: Instalación JDK de Java

a. Comprobar la versión de java, utilizando el siguiente comando:

$ java -version

b. Si la versión no es la 1.7 descargar la versión de java adecuada de su página web. En nuestro caso sería, ir al siguiente enlace:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

b1.Buscar la versión Java SE 7 Update 6 JDK, que encontraremos aquí:

http://www.oracle.com/technetwork/java/javase/downloads/jdk6-downloads-1637591.html

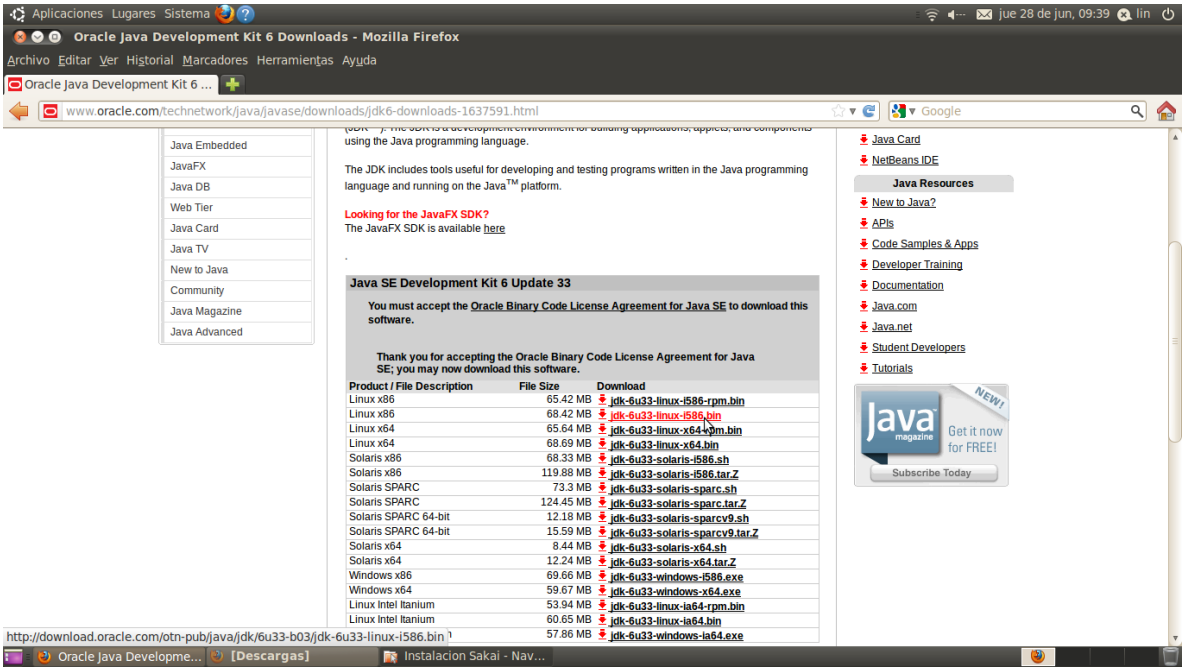

Figura 2: JDK de JAVA

b2. Descargar el jdk que se adapte a tu sistema operativo, en nuestro caso como estamos trabajando con Linux x86 nos descargaríamos la versión *jdk-7u6-linux-i586.bin*.

c. Ir a la carpeta donde se ha descargado, y ejecutarlo, para ello debemos darle permisos que se haría:

$chmod +x *jdk-7u6-linux-i586.bin*

c.1 Ahora lo ejecutaríamos:

$ *./jdk-7u6-linux-i586.bin*

Con lo que instalaría el jdk i dejaría un un nuevo directorio extraído.

d. Mover el directorio extraído a /etc para trabajar con más facilidad:

$ mv *jdk-7u6-linux-i586 /etc/jdk* (aquí renombramos el directorio como jdk)

e. A continuación habrá que establecer las variables JAVA_HOME, y JAVA_OPTS. Para ello nos movemos a nuestro directorio home:

$ *cd ~*

Y abrimos el fichero .bashrc que es donde vamos a declarar las variables:

$ *gedit .bashrc*

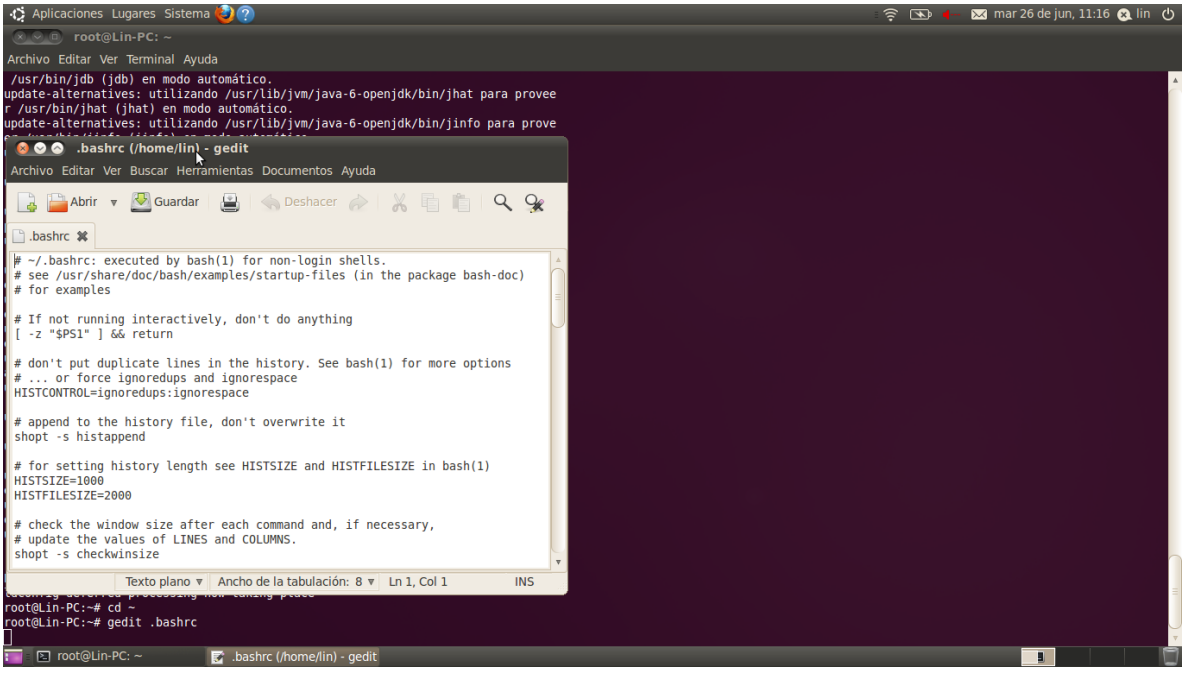

Figura 3: Variables .bashrc

Para declarar las variables habrá que añadirlas al fichero de la siguiente forma:

declare -x JAVA_HOME="/etc/jdk" (La ruta donde tenemos instalado el jdk)

declare -x JAVA_OPTS="-server -Xmx1028m -XX:MaxPermSize=320m -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true"

La variable JAVA_OPTS será las opciones que encontraremos dentro del fichero setenv el cual crearemos en el punto 6.

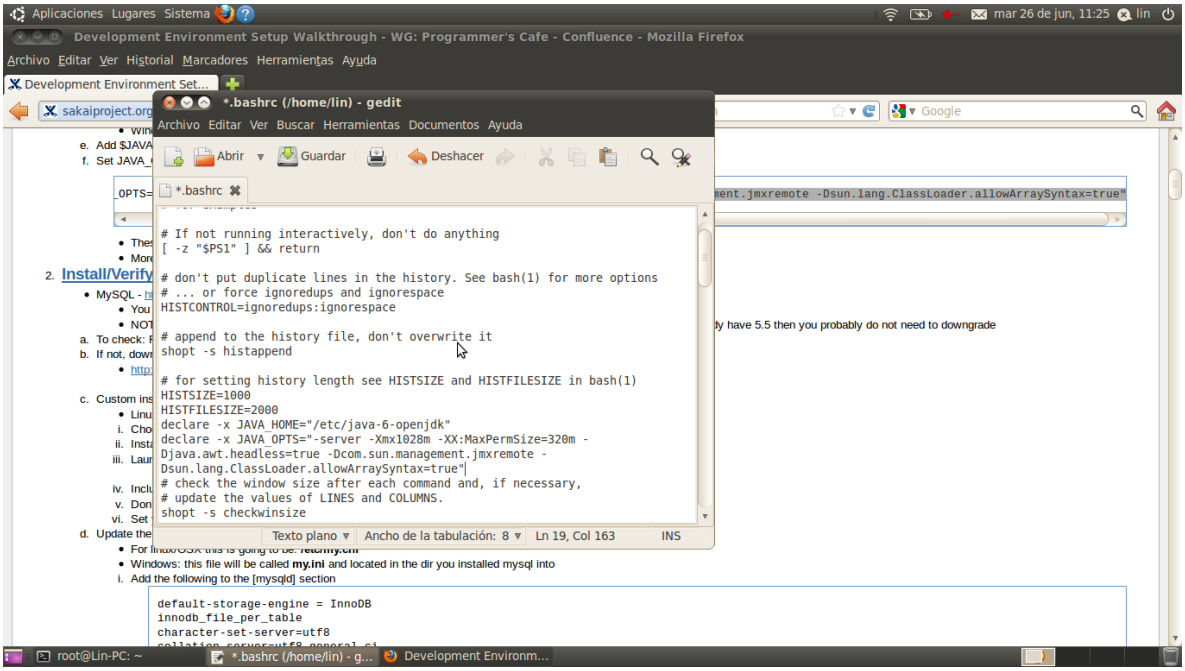

Figura 4: Variables JAVA

f. También habrá que añadir la ruta donde se encuentra el directorio bin del jdk que hemos instalado manualmente. Para ello, dentro del .bashrc a continuación de las últimas variables declaradas, escribimos:

export PATH=$JAVA_HOME/bin:$PATH

Con lo cual ya tendremos todas las variables declaradas, por lo que guardamos el fichero, lo cerramos y ejecutamos el siguiente comando en consola para que se actualice:

$ source .bashrc (tendremos que estar en el directorio home para poder ejecutarlo)

Paso 2: Instalar MySQL 5.1

a. Comprobar si tienes instalado mysql, para ello ejecutamos el comando:

$ mysql -help

En caso afirmativo si la versión instalada es distinta de la 5.1, instalar la 5.1, si es negativo vamos a descargarlo.

b. Para poder instalar MySQL, utilizaremos el apt-get porque es la forma más rápida, el comando a introducir sería:

$ apt-get install mysql-server-5.1

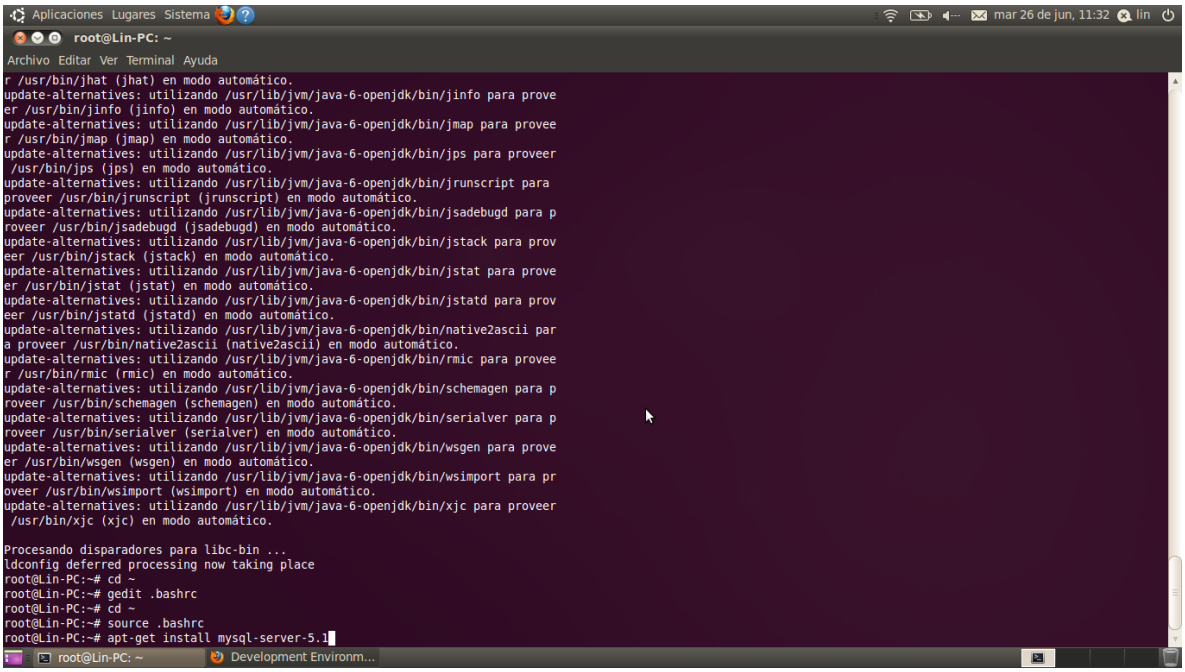

Figura 5: Instalación mysql

Si todo va bien, introduciremos la contraseña que deseamos que tenga, en nuestro caso será 'a', y con esto ya tendremos instalado el MySQL.

Paso 3: Crear la base de datos de Sakai

a. Para crear la base de datos de Sakai debemos ejectuar el comando siguiente:

$ mysql -uroot -p (Introducir la contraseña que hemos puesto al instalar mysql)

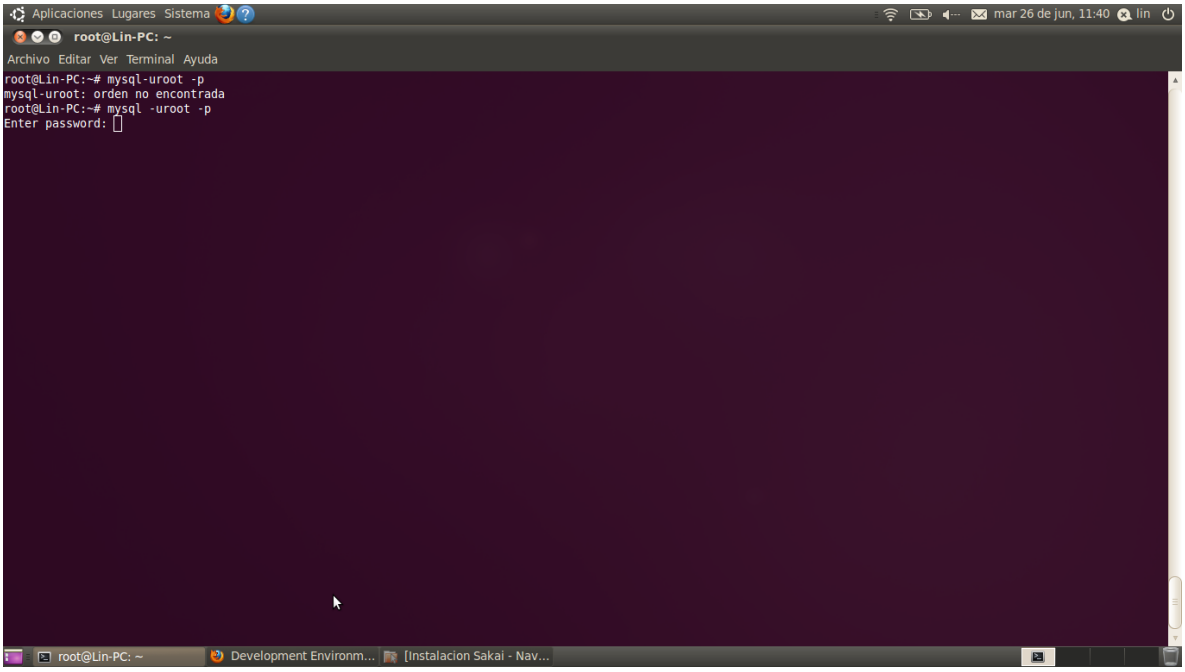

Figura 6: Contraseña mysql

Introducimos la contraseña, y en la terminal escribimos lo siguiente:

mysql> create database Sakai default character set utf8;

mysql> grant all privileges on Sakai. to 'Sakai'@'localhost' identified by 'a';*

mysql> flush privileges;

mysql> quit

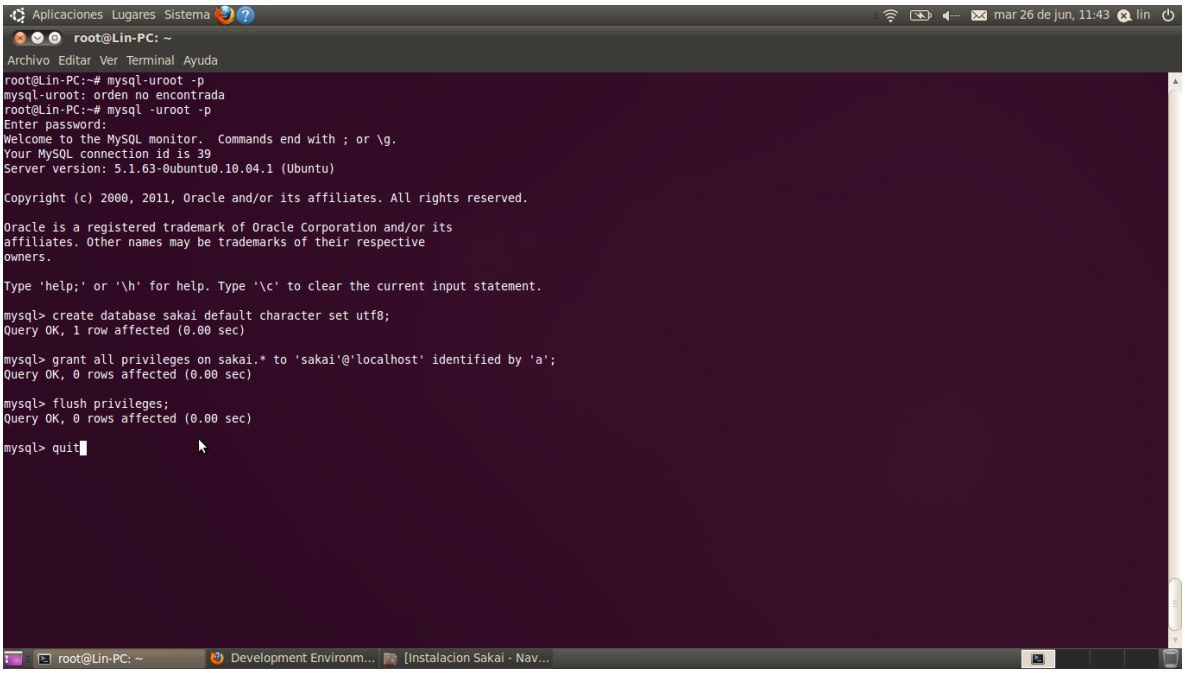

Figura 7: Base de Datos

Y con esto ya tenemos la base de datos creada.

Paso 4: Descargar e Instalar Maven2.2+

a. Descargar la versión 2.2.X de Maven e instalarla, para ello utilizaremos el apt-get:

$ apt-get install maven2

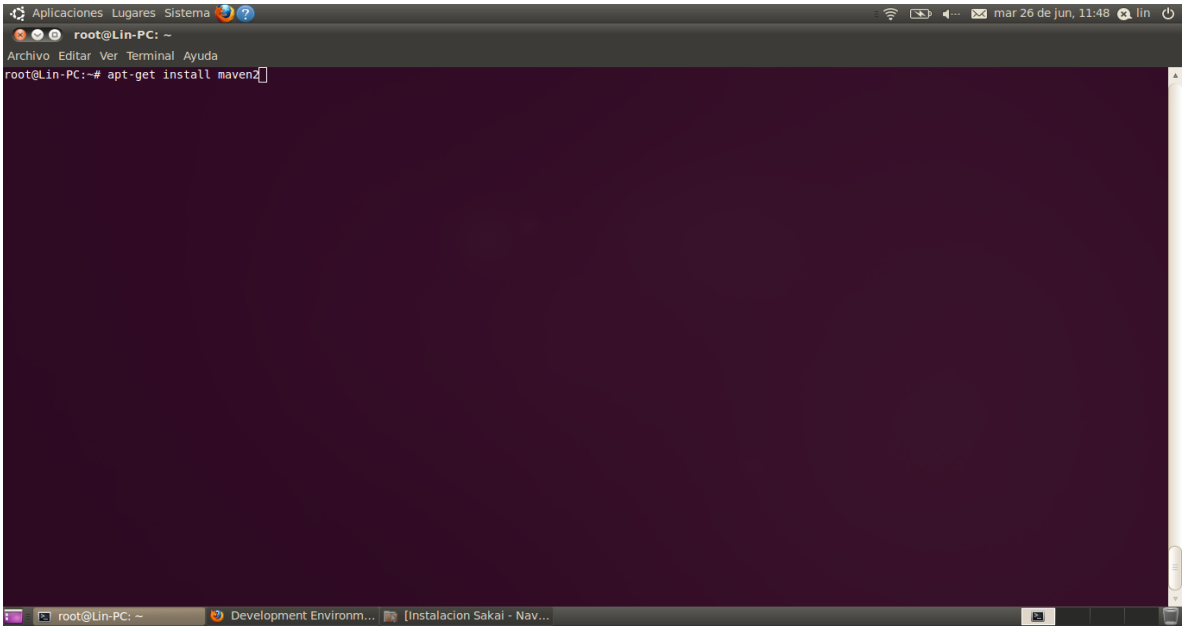

Figura 8: Instalar maven2

b. Declarar las variables MAVEN2_HOME y MAVEN2_OPTS en .bashrc, para ello iremos a nuestro home y abriremos el archivo .bashrc:

$ cd ~

$ gedit .bashrc

Ahora declaramos las variables de la siguiente manera:

declare -x MAVEN2_HOME="/etc/maven2" *(Es donde el apt-get instala Maven)*

declare -x MAVEN_OPTS="-Xms128m -Xmx796m -XX:PermSize=64m -XX:MaxPermSize=172m"

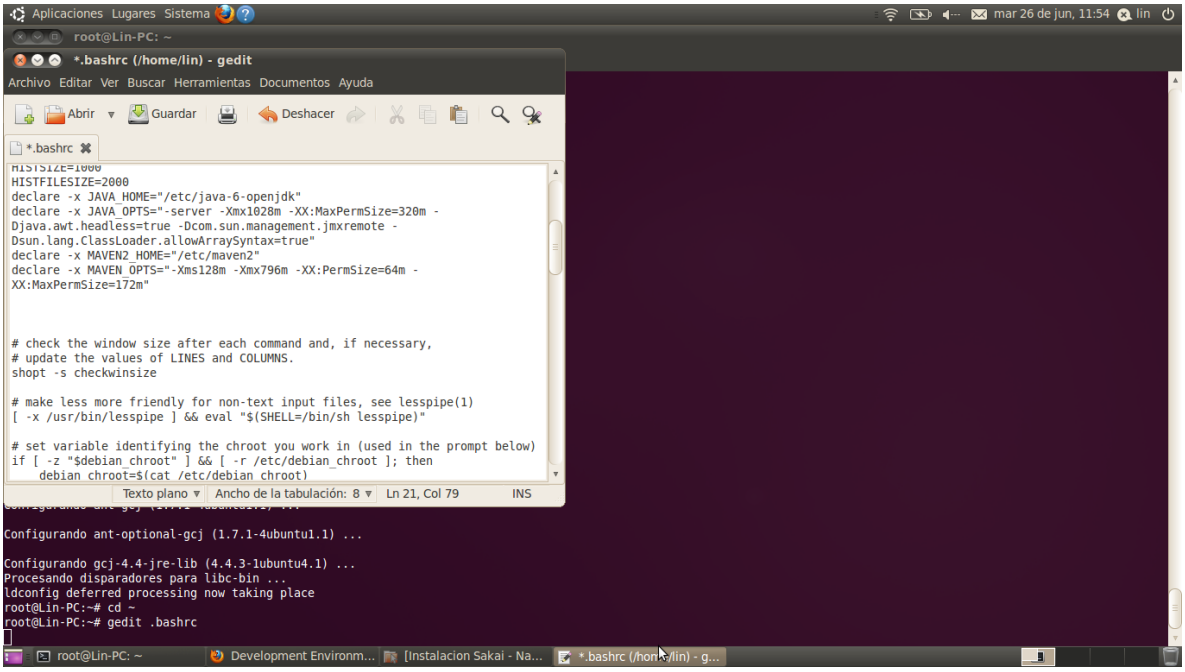

Figura 9: Variables maven2

Guardamos el archivo, y ejecutamos el siguiente comando para que se actualice:

$ source .bashrc

Paso 5: Instalamos subversión

a. Si no tenemos instalado subversión, la forma más sencilla de hacerlo es utilizando el apt-get, para ello ejecutaremos en la línea de comandos:

$ apt-get install subversión

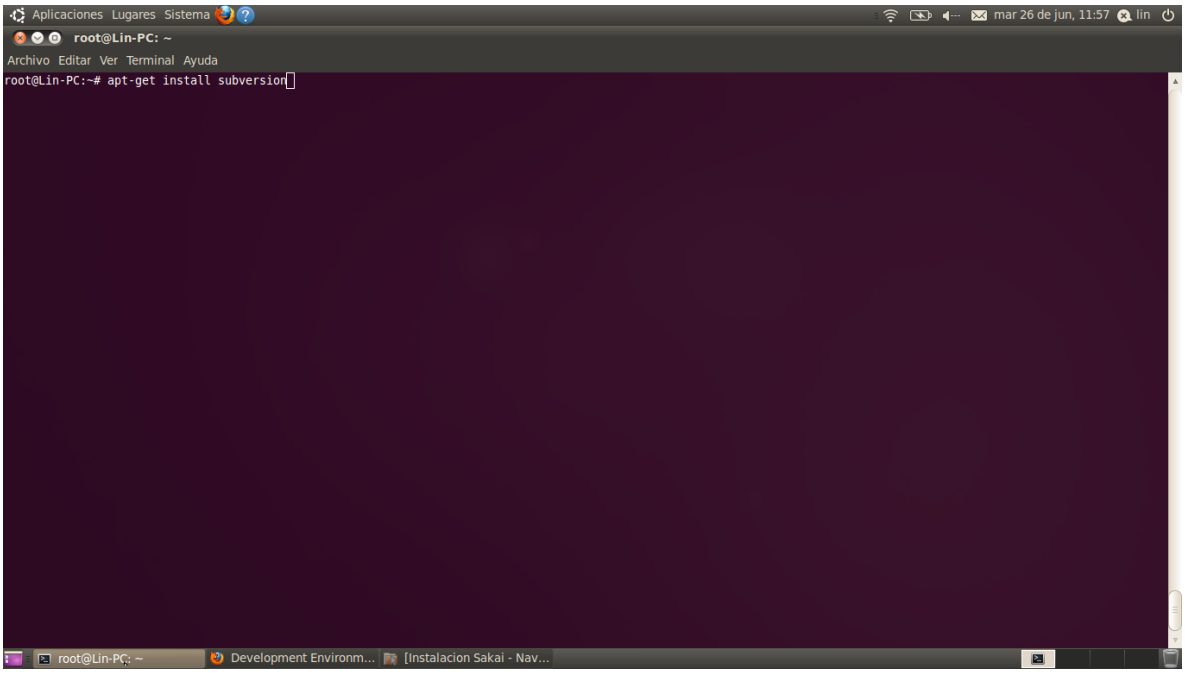

Figura 10: Instalación Subversion

Y con esto tendremos instalado subversión.

b. Crear variables de entorno de subversión. Será SUBVERSION_HOME. Para crear la variable hay que hacer lo mismo que en los pasos anteriores, ir al home, abrir el .bashrc, declarar la variable, guardar el fichero y actualizarlo. Estos pasos podemos verlos a continuación:

$ cd ~

$ gedit .bashrc

declare -x SUBVERSION_HOME="/etc/subversion"

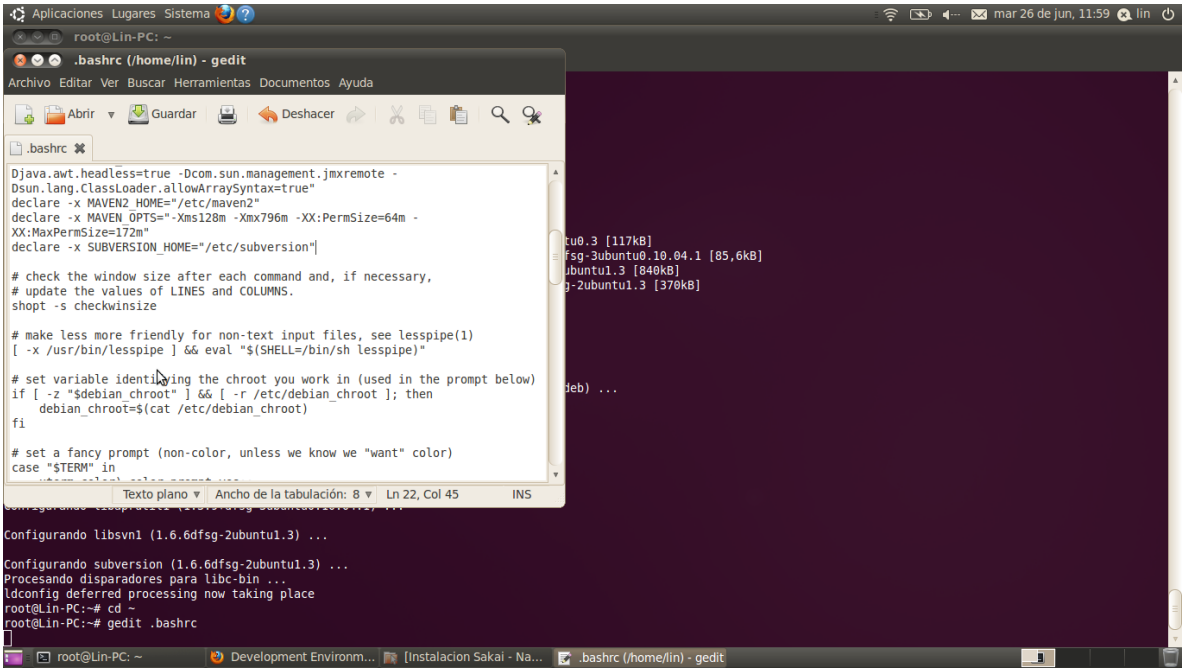

Figura 11: Variables Subversion

Ahora guardamos el archivo y lo actualizamos con:

$ source .bashrc

Con esto hecho ya habríamos terminado de instalar Subversion.

Paso 6: Descargar e Instalar Tomcat 7.0.21+

a. Para descargar Tomcat debemos ir a su página de descarga, que es:

http://tomcat.apache.org/download-70.cgi

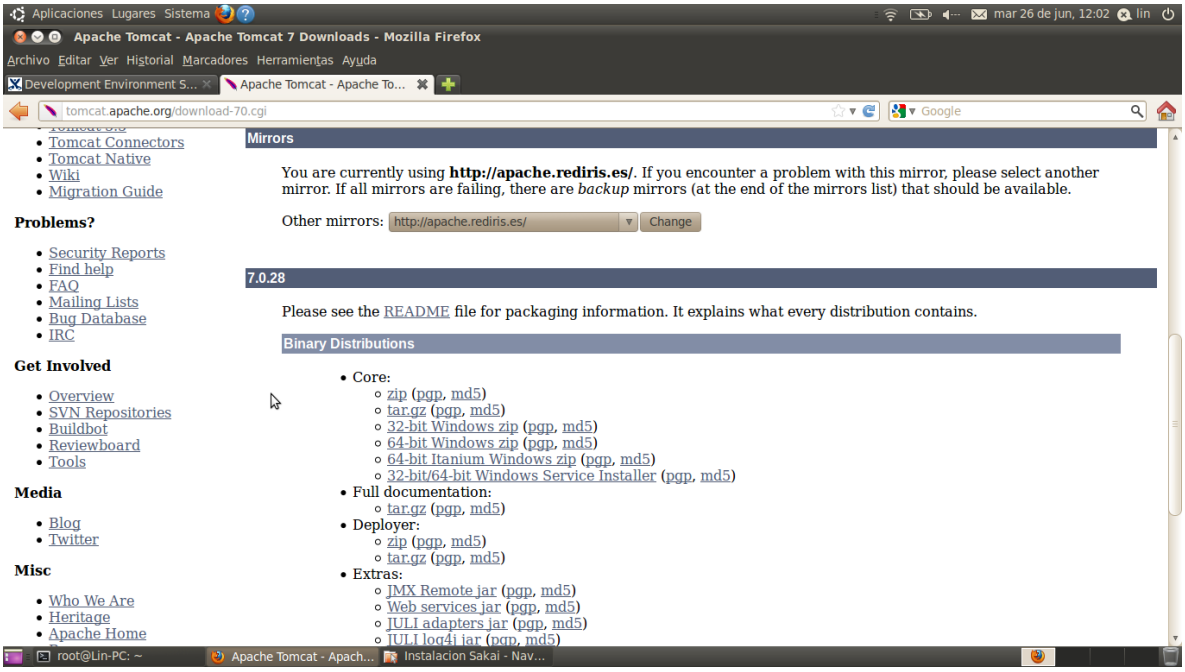

Figura 12:Instalación Tomcat

Dentro de esta página debemos de descargarnos el archivo que corresponda a nuestro sistema operativo. En nuestro caso nos descargamos el .tar.gz.

b.Vamos a la carpeta /opt y le damos permisos para que podamos acceder a ella sin restricciones. Movemos el fichero descargado a opt y lo extraemos de la siguiente forma:

$ tar -xzvf apache-tomcat-7.0.25.tar.gz

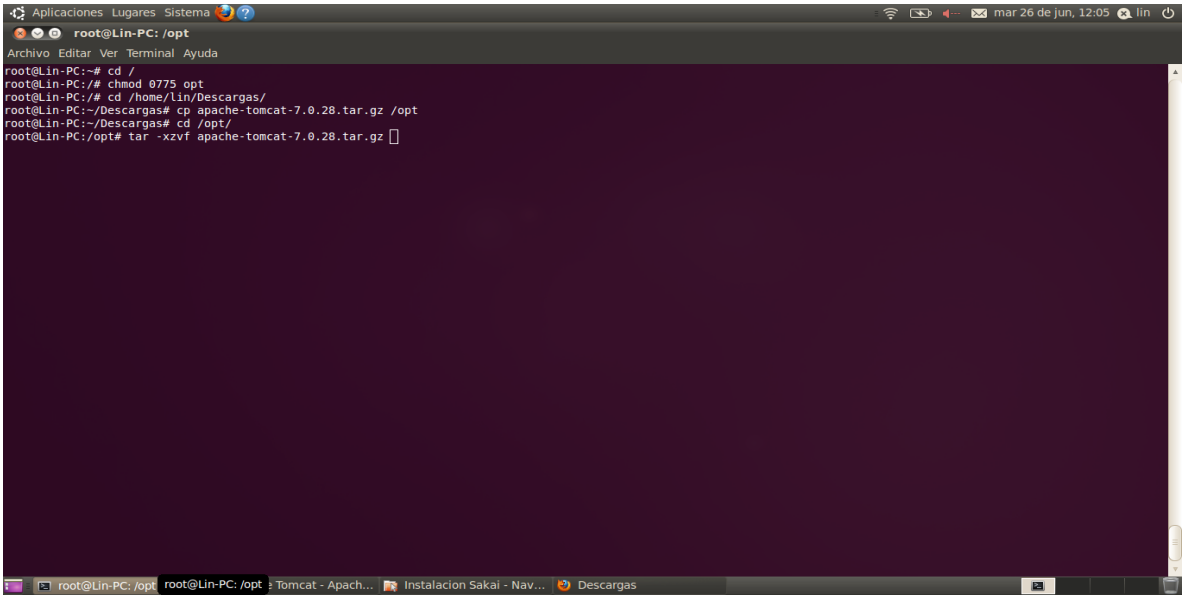

Figura 13: Descompresión de tomcat

$ mv apache-tomcat-7.0.25 /opt/tomcat (aquí lo renombramos a Tomcat)

c. Vamos al directorio conf dentro de Tomcat, que lo podemos hacer así:

$ cd /opt/tomcat/conf

Y editamos el fichero server.xml, en el que deberemos añadir URIEncoding="UTF-8" en la parte del puerto de conexión 8080, haríamos algo como esto:

$ gedit server.xml

...

<Connector port="8080" URIEncoding="UTF-8"...

...

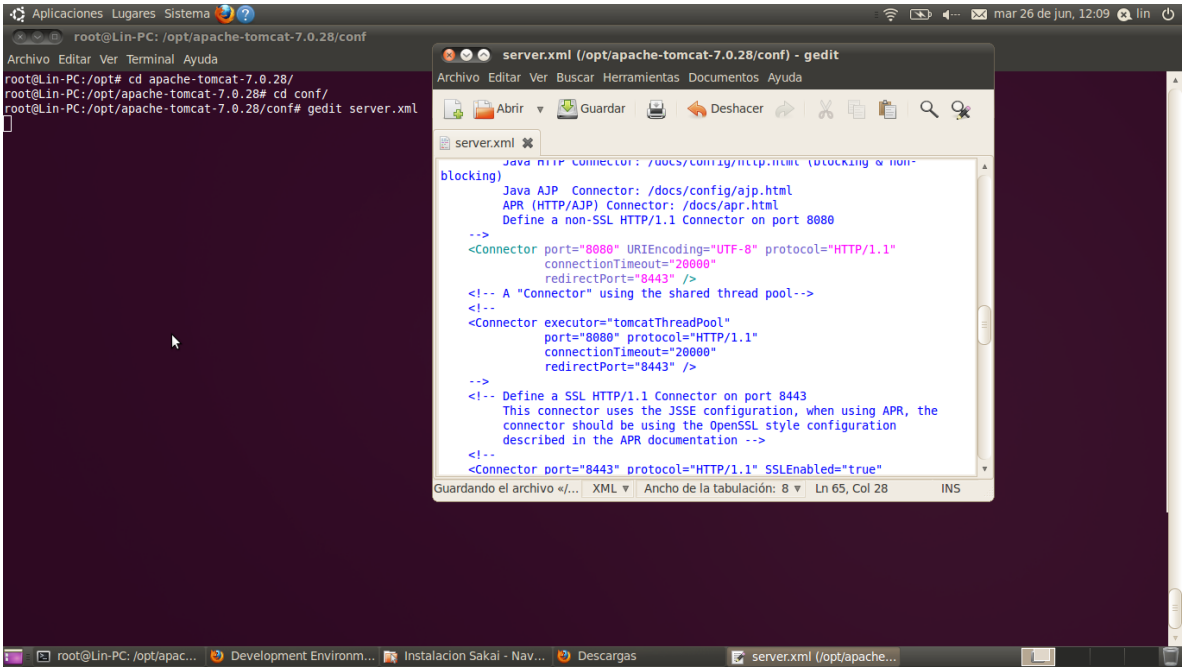

Figura 14: Editar archivo server.xml

Guardaríamos el fichero, y tendríamos que declarar las variables de entorno.

d. En este caso solo hay una variable de entorno que es CATALINA_HOME, haremos lo mismo que en los pasos anteriores:

$ cd ~

$ gedit .bashrc

declare -x CATALINA_HOME="/opt/tomcat"

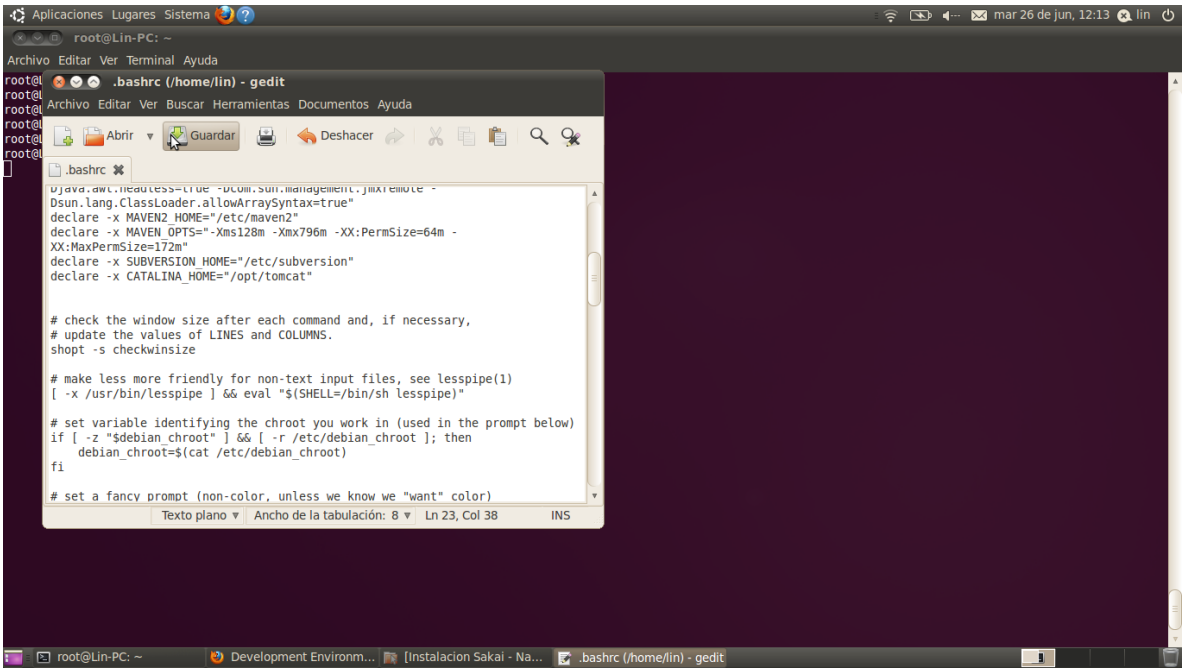

Figura 15: Variables tomcat

Guardamos el fichero y lo actualizamos:

$ source .bashrc

e. Ahora vamos a crear el fichero seteven.sh que contendrá el JAVA_OPTS como hemos comentado en el paso 1.

Primero iremos al directorio bin de Tomcat:

$ cd /opt/tomcat/bin

Crearemos el fichero:

$ gedit setenv.sh

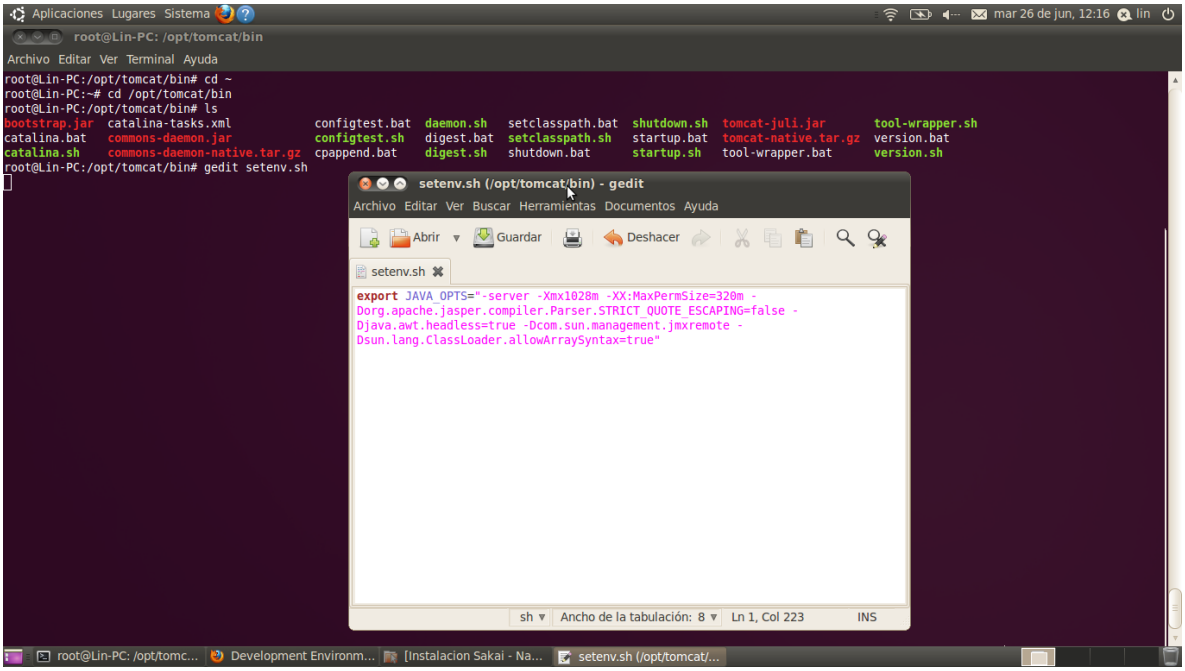

Figura 16: Creación setenv.sh

Y le añadiremos:

> export JAVA_OPTS= "-server -Xmx1028m -XX:MaxPermSize=320m -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true"

Guardamos el fichero y ya tenemos el setenv.sh creado.

f. Ahora modificaremos el fichero catalina.properties que se encuentra en el directorio conf de Tomcat:

$ cd /opt/tomcat/conf

$ gedit catalina.properties

Y le añadiremos a la línea que empieza por "common.loader=...":

,${catalina.base}/common/classes/,${catalina.base}/common/lib/.jar*

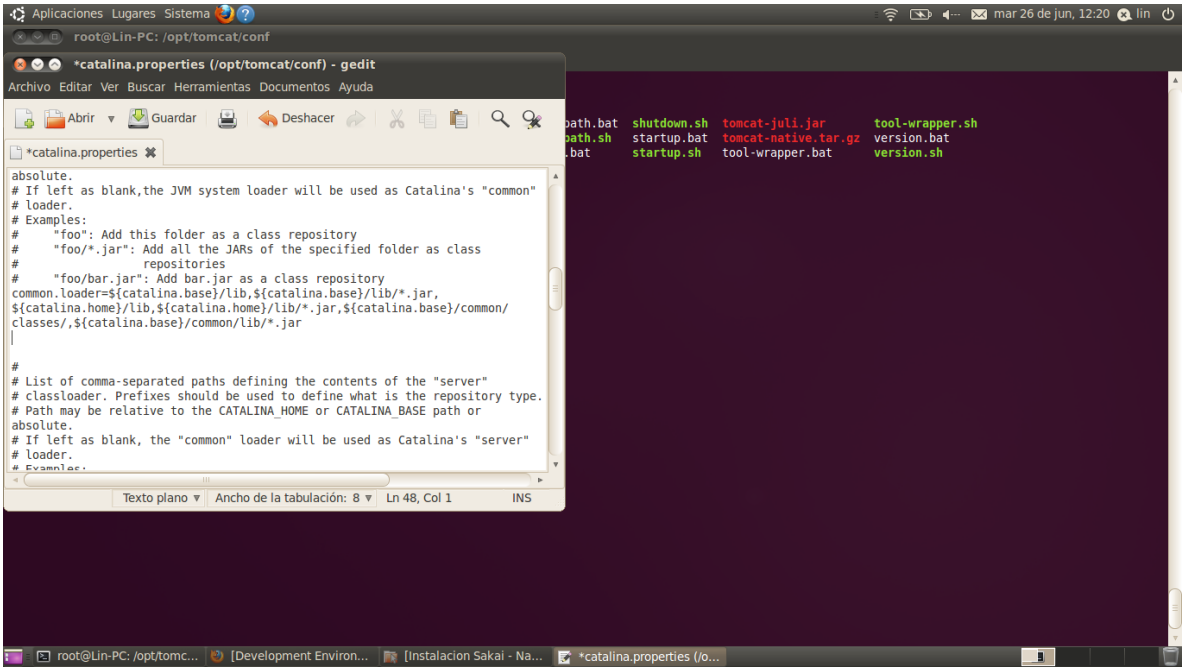

Figura 17: Edición catalina.properties common.loader

A la línea que empieza por "shared.loader=...":

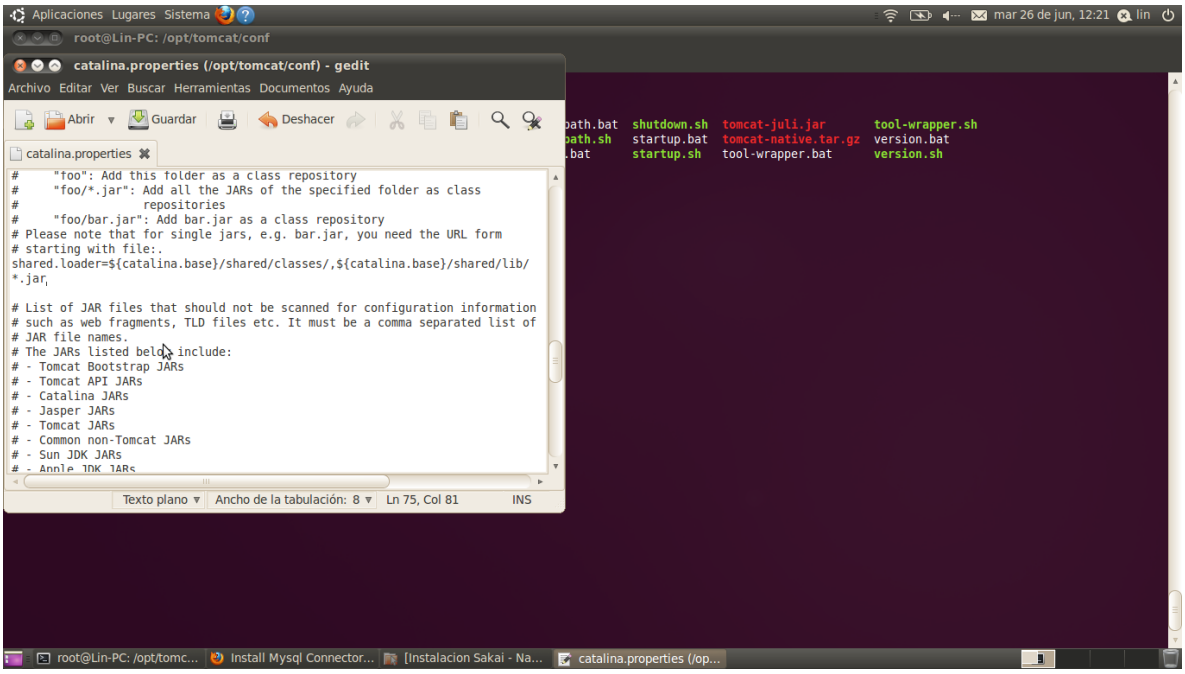

Figura 18: Edición catalina.properties shared.loader

Y a la que empieza por "server.loader=..."

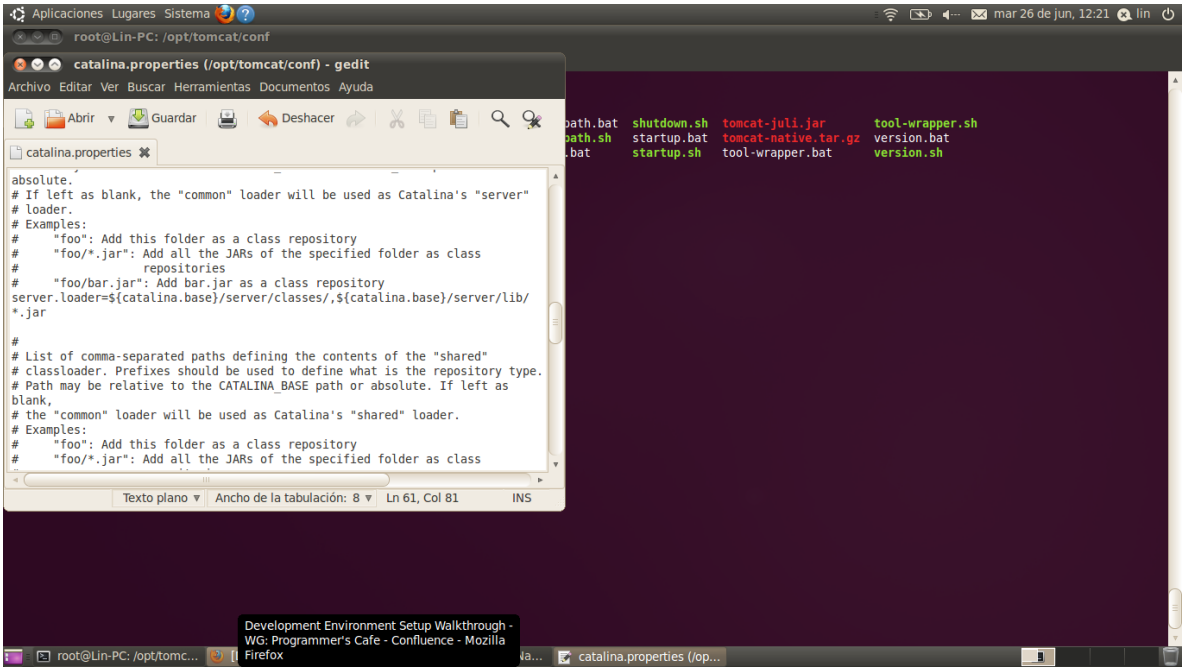

Figura 19: Edición catalina.properties server.loader

${catalina.base}/server/classes/,${catalina.base}/server/lib/.jar*

Ahora guardamos el archivo para finalizar este paso.

Paso7: Descargar e instalar MySQL Connector

a. Como estamos ejecutando una versión de MySQL superior al MySQL 5.1 (concretamente MySQL 5.5), descargamos el conector del siguiente enlace:

http://www.mysql.com/downloads/connector/j/

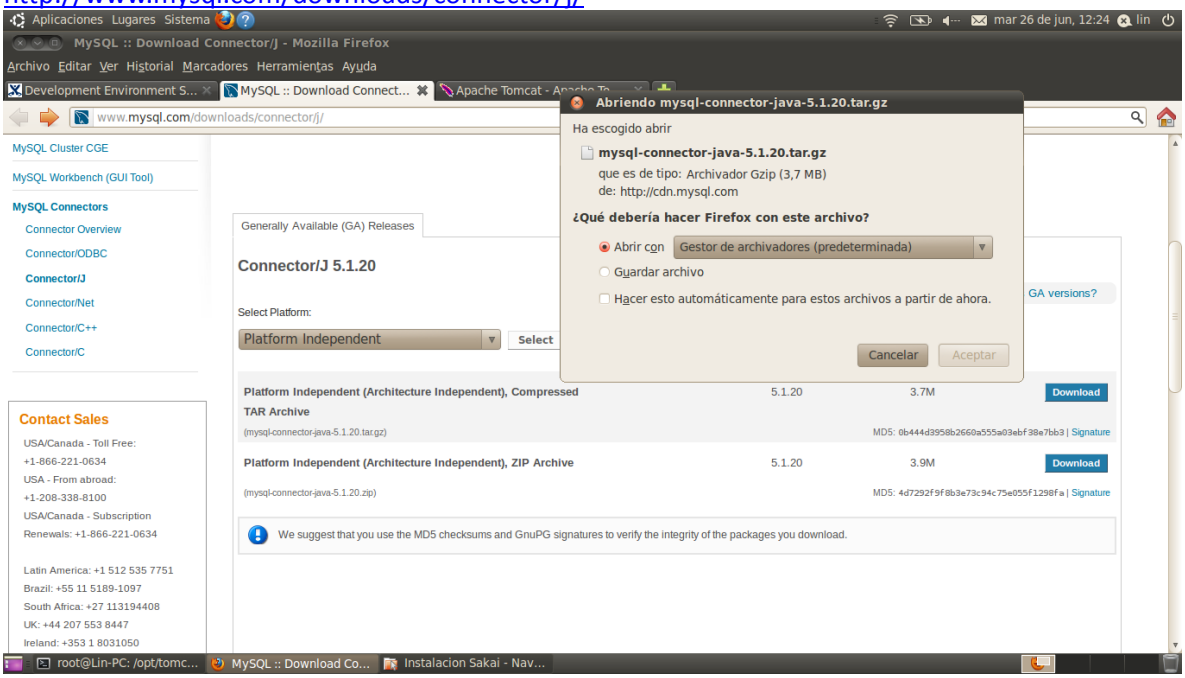

Figura 20: Descargar MySql Connector

b. Extraemos el archivo descargado en /opt

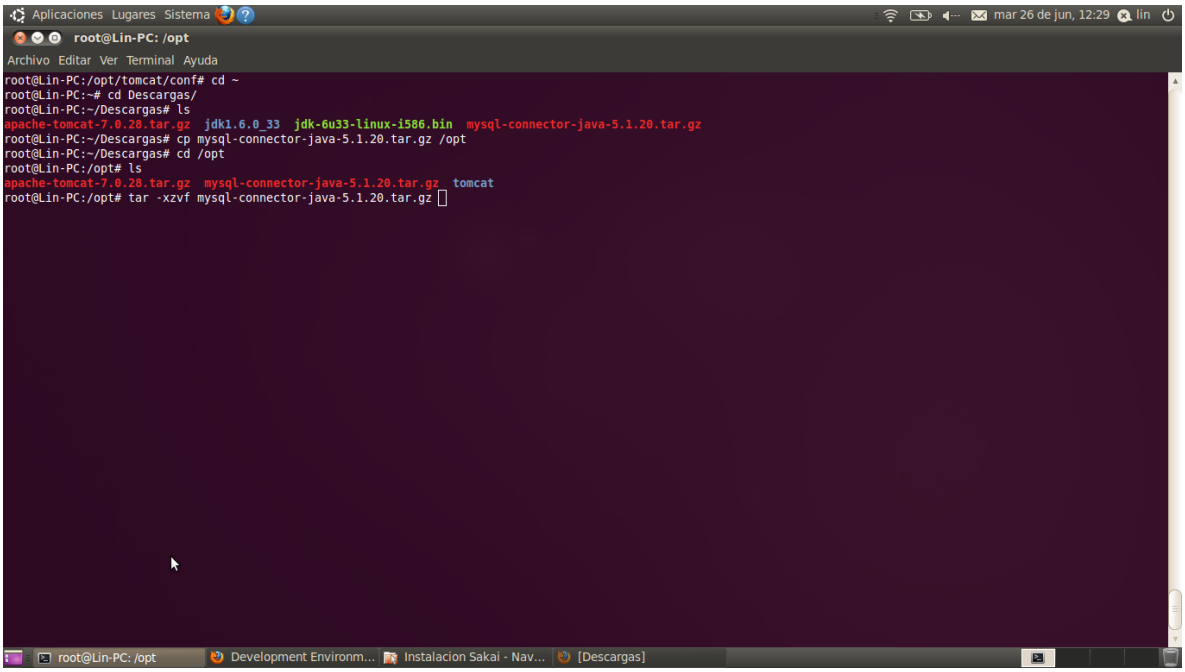

Figura 21: Extraer MySql Connector

c. Copiamos mysql-connector-java-<version>-bin.jar en $CATALINA_HOME/lib (que en nuestro caso sería en /opt/tomcat/lib).

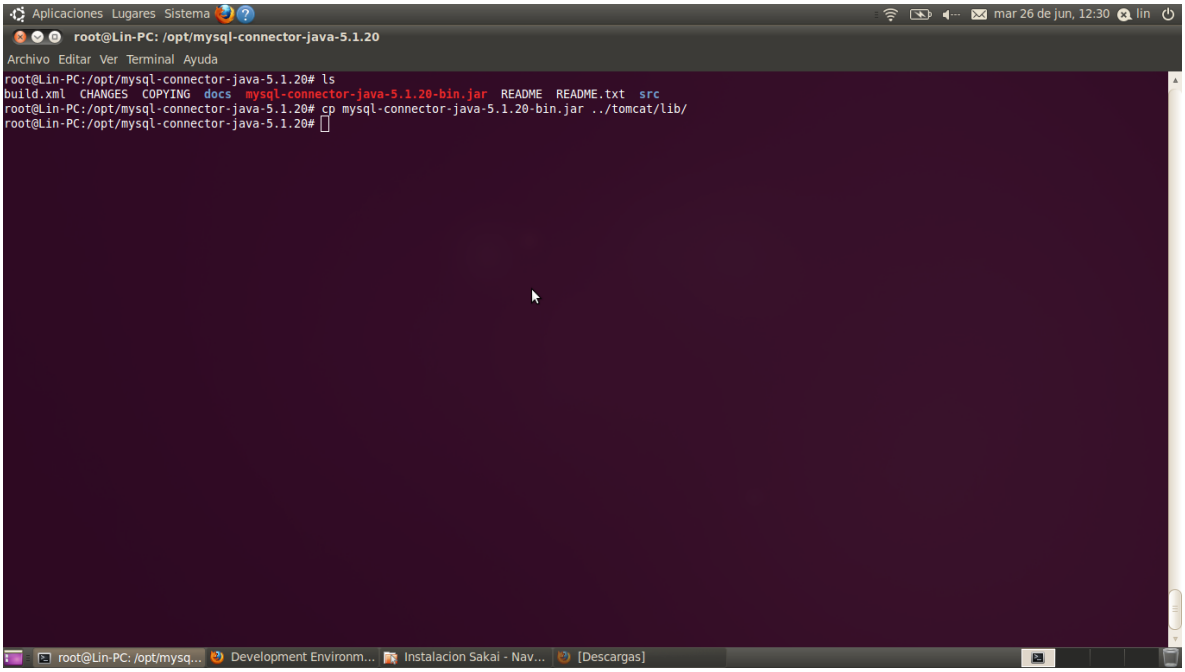

Figura 22:Copiar MySql Connector en $CATALINA_HOME

d. Borrar el archivo extraído.

Paso 8: Utilizar Subversión para descargar el código de Sakai

a. Cambiar al directorio /opt

b. Utilizar el comando:

svn checkout https://source.Sakaiproject.org/svn/Sakai/branches/Sakai-2.8.x/ Sakai-src

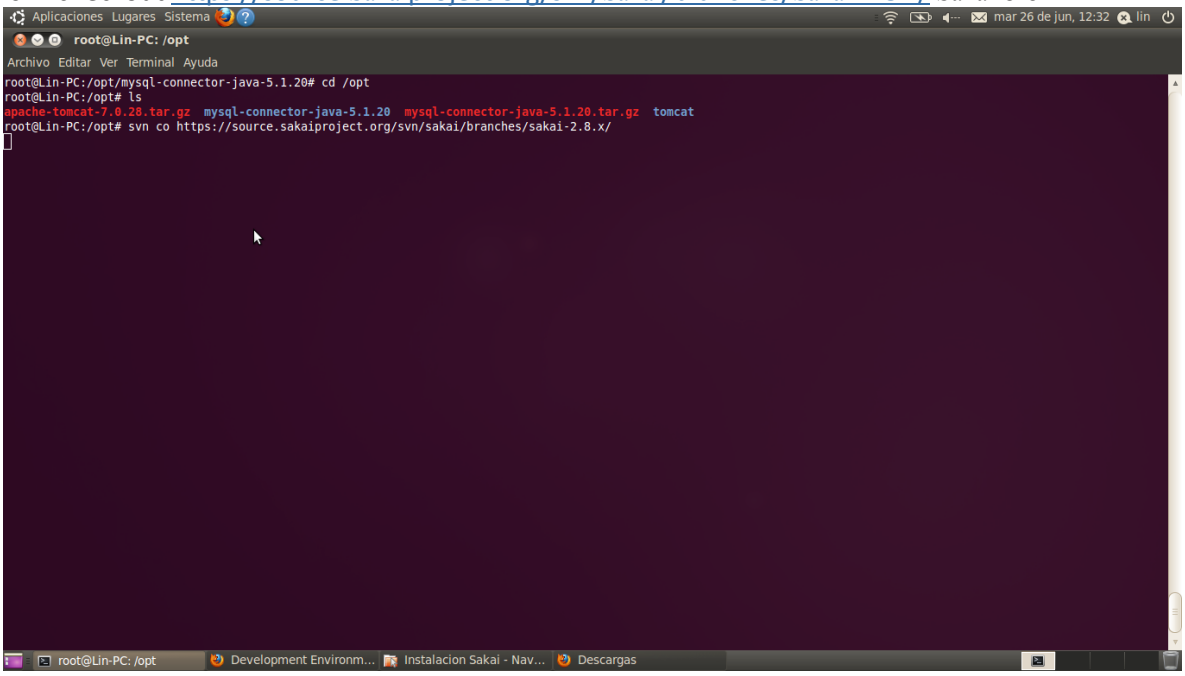

Figura 23: Descargar repositorios Sakai

Con este comando descargamos los repositorios de Sakai a un nuevo directorio que en este caso sería Sakai-src. Como se puede ver en el enlace estamos descargando los repositorios de la versión 2.8 de Sakai. En caso de requerir otra versión solo se tendría que cambiar el enlace.
Este proceso puede tardar entre 5 y 10 minutos.

Paso 9: Crear el archivo Sakai.properties

a. Vamos a la dirección $CATALINA_HOME y creamos un directorio denominado Sakai.

b. Copiar el archivo default.Sakai.properties que se encuentra en la ruta <Sakai-src>/config/configuration/bundles/src/bundle/org/Sakaiproject/config/bundle/default.Sakai.properties a $CATALINA_HOME/Sakai/Sakai.properties. Al copiarlo en esta ruta también renombramos el fichero a Sakai.properties.

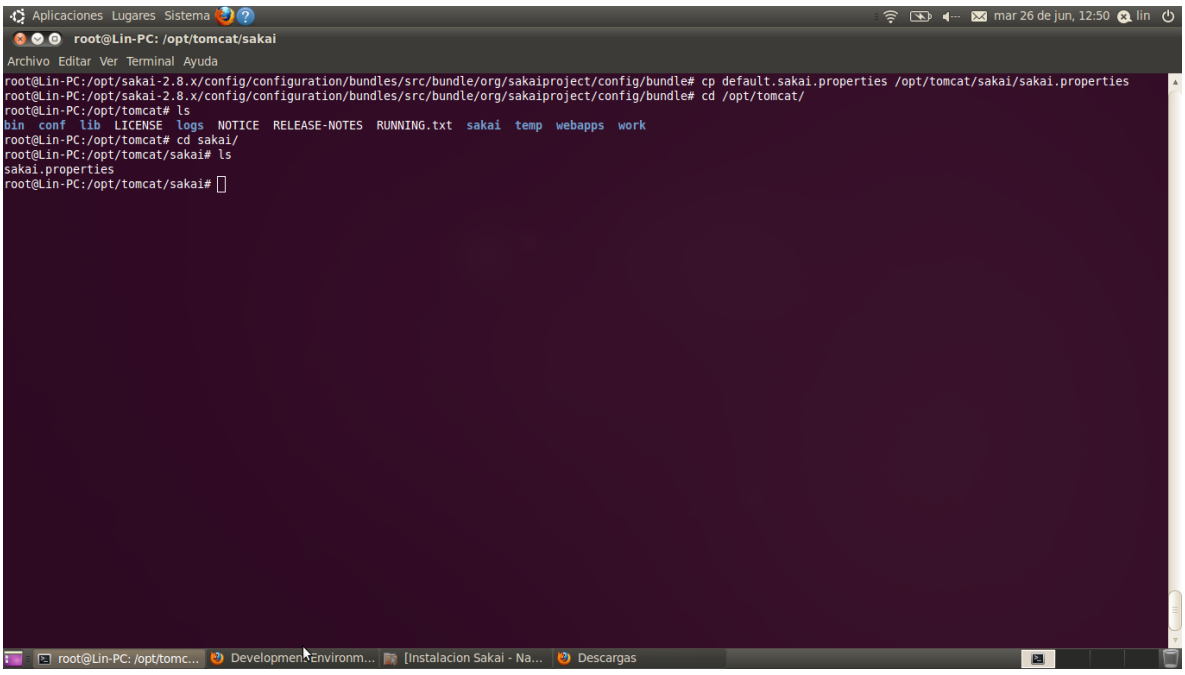

Figura 24: Copiar default.Sakai.properties a Sakai.properties

c. Editamos el archivo Sakai.properties, para ello nos dirigiremos a la sección marcada como #DATABASE:

c1. Estableceremos el nombre de usuario de la base de datos:
username@javax.sql.BaseDataSource=Sakai

c2. Estableceremos la contraseña de la base de datos:
password@javax.sql.BaseDataSource=ironchef

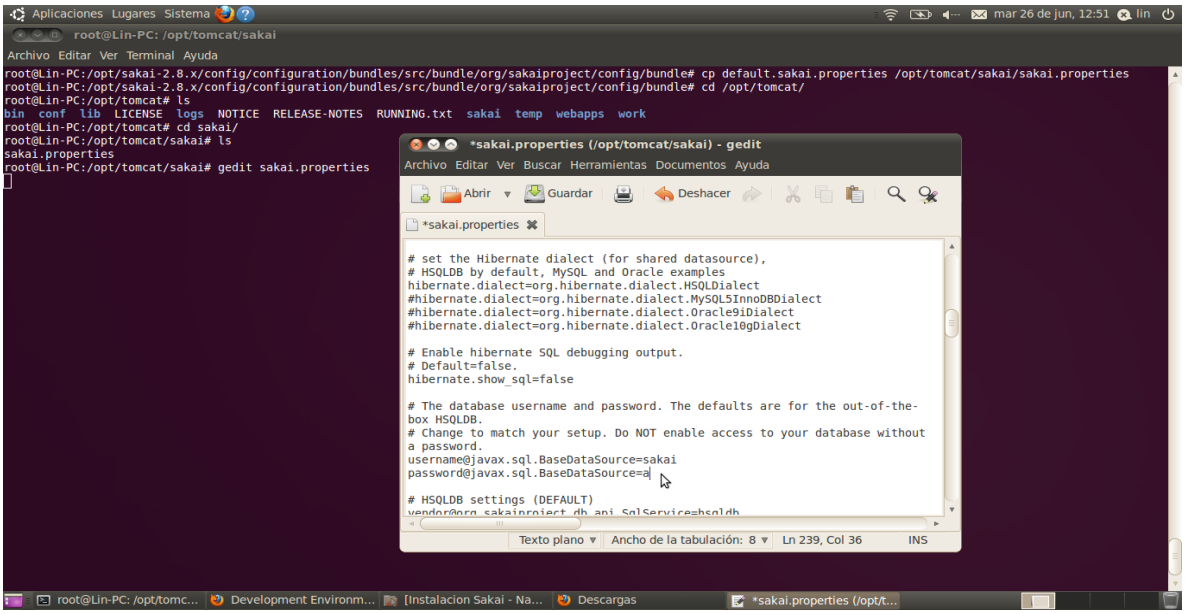

Figura 25: Establecer usuario y contraseña de la Base de datos

c3. Ahora nos dirigiremos a la sección **# HSQLDB settings** y comentaremos las siguientes líneas:

HSQLDB settings (DEFAULT)
#vendor@org.Sakaiproject.db.api.SqlService=hsqldb
#driverClassName@javax.sql.BaseDataSource=org.hsqldb.jdbcDriver
#hibernate.dialect=org.hibernate.dialect.HSQLDialect
#validationQuery@javax.sql.BaseDataSource=select 1 from INFORMATION_SCHEMA.SYSTEM_USERS
Two hsqldb storage options: first for in-memory (no persistence between runs), second for disk based.
#url@javax.sql.BaseDataSource=jdbc:hsqldb:mem:Sakai
#url@javax.sql.BaseDataSource=jdbc:hsqldb:file:${Sakai.home}db/Sakai.db

c4. A continuación nos situaremos en la sección **# MySQL settings** y comentaremos estas líneas:

MySQL settings
vendor@org.Sakaiproject.db.api.SqlService=mysql
driverClassName@javax.sql.BaseDataSource=com.mysql.jdbc.Driver
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
url@javax.sql.BaseDataSource=jdbc:mysql://127.0.0.1:3306/Sakai?useUnicode=true&characterEncoding=UTF-8
validationQuery@javax.sql.BaseDataSource=select 1 from DUAL
defaultTransactionIsolationString@javax.sql.BaseDataSource=TRANSACTION_READ_COMMITTED

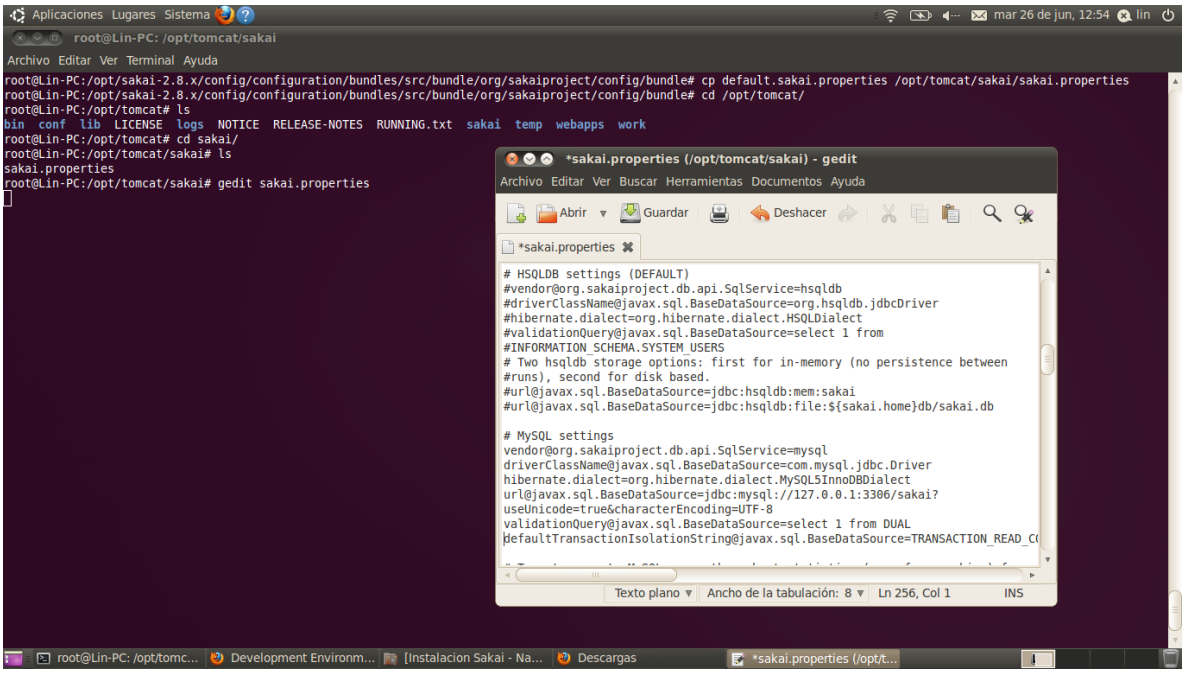

Figura 26: Modificar secciones Base de Datos

d. Finalmente guardaremos los cambios del fichero Sakai.properties.

Paso 10: Crear el archivo settings.xml de Maven

a. Crear un nuevo archivo xml en el directorio .m2 del home de tu usuario que se llamará settings.xml.

b. En este archivo agregamos las siguientes líneas:

```
<settings xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                http://maven.apache.org/xsd/settings-1.0.0.xsd">
 <profiles>
  <profile>
   <id>tomcat5x</id>
   <activation>
    <activeByDefault>true</activeByDefault>
   </activation>
   <properties>
    <appserver.id>tomcat5x</appserver.id>
    <appserver.home>/opt/tomcat</appserver.home>
    <maven.tomcat.home>/opt/tomcat</maven.tomcat.home>
    <Sakai.appserver.home>/opt/tomcat</Sakai.appserver.home>
    <surefire.reportFormat>plain</surefire.reportFormat>
    <surefire.useFile>false</surefire.useFile>
   </properties>
  </profile>
 </profiles>
</settings>
```

Listado 3. Creación archive settings.xml

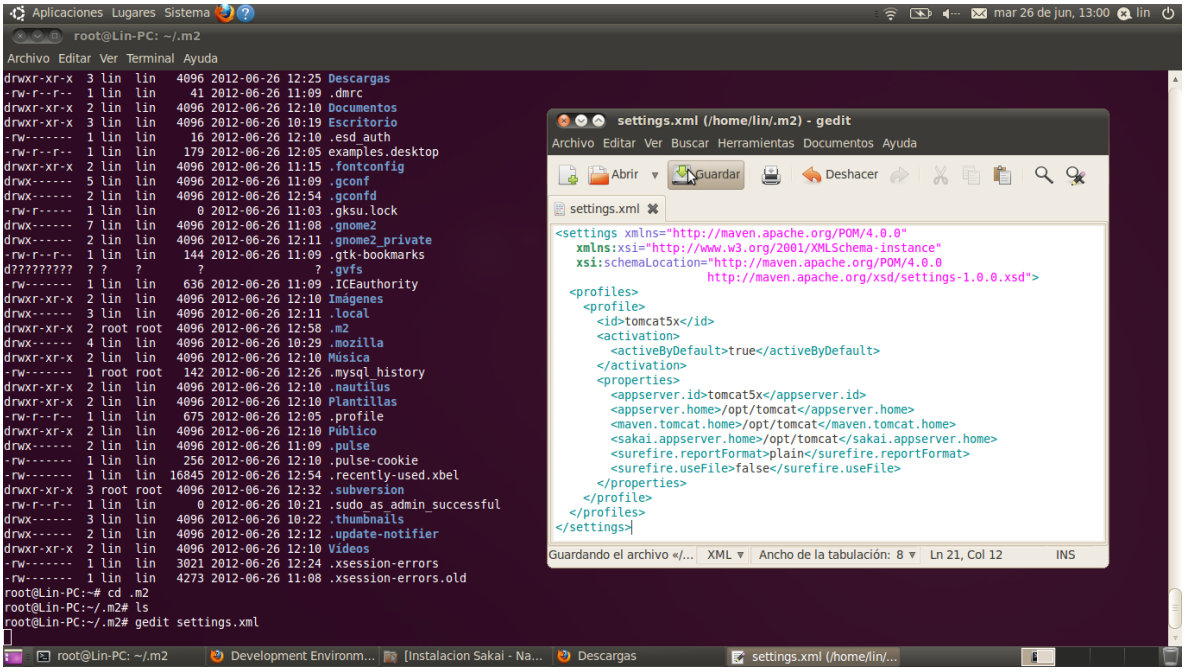

Figura 27: Crear settings.xml

c. Finalmente guardaremos y cerraremos el archivo.

Paso 11: Utilización de Maven para construir Sakai

a. Ir al directorio /opt/<Sakai-src>

b. Ejecutar el comando mvn -Pcafe clean install para construir la versión más simple de Sakai utilizando Maven. Este proceso puede tardar entre 5 y 10 minutos.

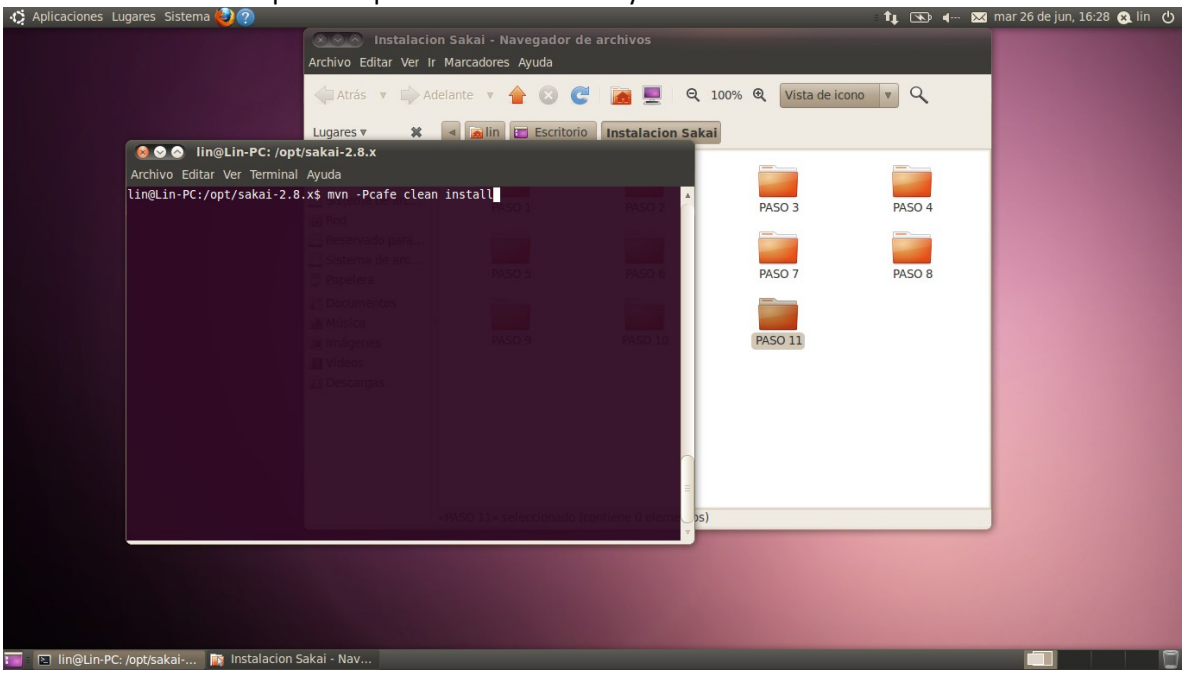

Figura 28: Construcción de Sakai con Maven

c. Ejecutar mvn -Pcafe Sakai:deploy para desplegar Sakai a tu Tomcat utilizando Maven.

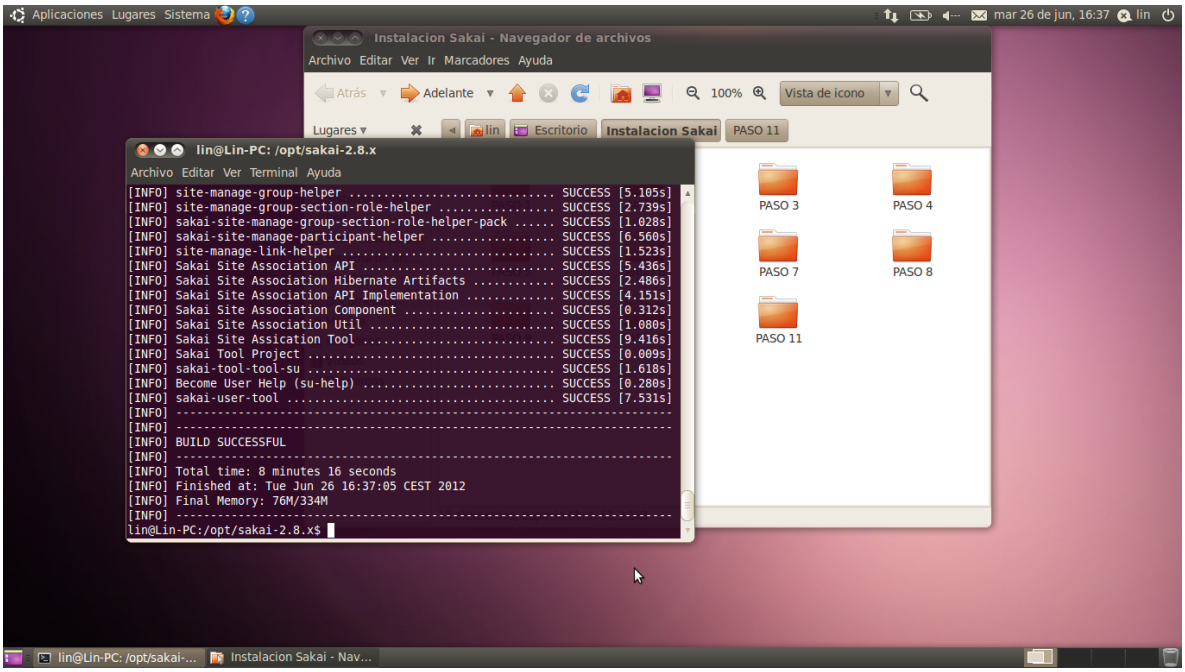

Figura 29: Despliegue de Sakai con maven

c1. Cuando termine si todo ha ido bien nos dará como resultado BUILD SUCCESSFUL

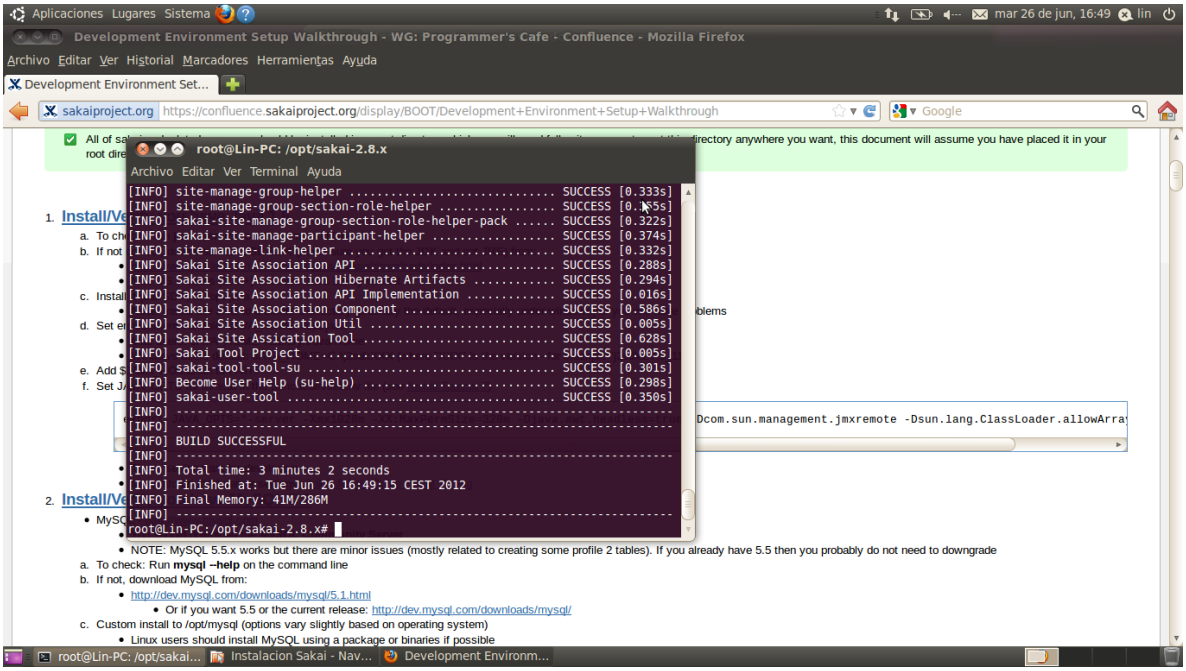

Figura 30: Construcción completa de Sakai

Si quieres construir una versión completa de Sakai solo tienes que omitir el -Pcafe del comando anterior.

Paso 12: Iniciar Tomcat y comprobar que Sakai se ejecuta adecuadamente

a. Inicar Tomcat ejecutando el archivo startup que se encuentra en $CATALINA_HOME/bin. Esperaremos más de un minuto para que el Tomcat arranque.

b. Abriremos un navegador y nos dirigiremos a la siguiente dirección http://localhost:8080/ para verificar que Tomcat está ejecutándose.

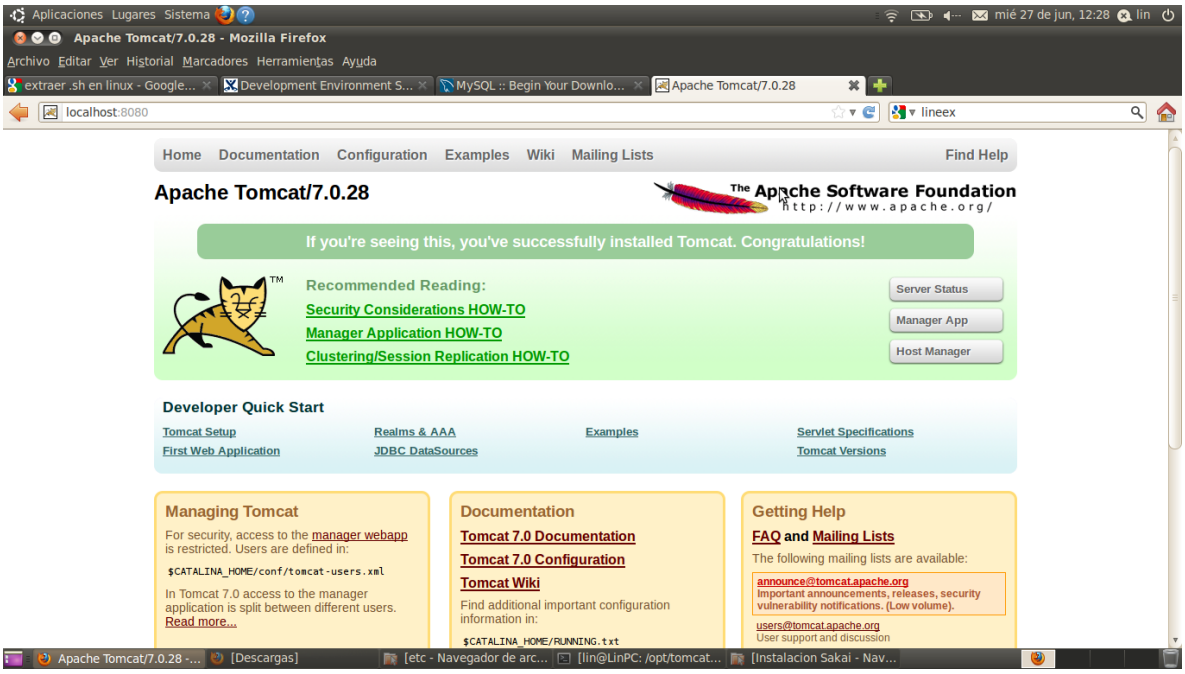

Figura 31: Iniciar Tomcat

c. Ahora comprobaremos que Sakai está ejecutándose dirigiéndonos a http://localhost:8080/portal

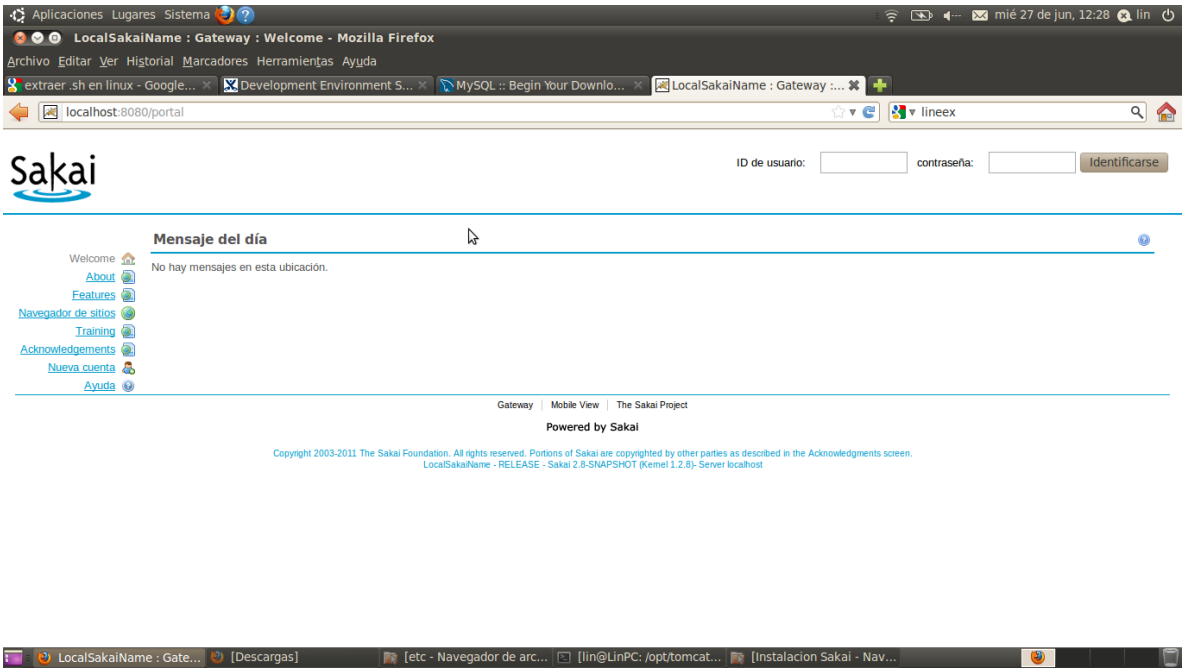

Figura 32: Iniciar Sakai

d. Nos loguearemos en Sakai utilizando el usuario admin y la contraseña admin.

Finalmente apagaremos tomcat utilizando el archivo shutdown que se encuentra en $CATALINA_HOME/bin .

CAPITULO 4: FUNCIONAMIENTO DEL SCRIPT

En este capítulo vamos a explicar cómo funcionaran los script que hemos desarrollado para el Sistema Operativo Linux concretamente para la distribución de Ubuntu con la versión 12.04,que es la versión LTS más actual.

Tenemos dos scripts instalacion1.sh y instalacion2.sh. Para que todo funcione primero ejecutamos el script instalacion1.sh después cerramos el terminal, abriremos otro nuevo y después de esto ejecutamos el script instalacion2.sh una vez termine esto si no ha habido ningún fallo inesperado debería funcionar todo correctamente.

Instalacion1.sh

```
# 1-Declararemos las variables de entorno.
mainDir=$HOME"/Archivos"
echo "declare -x JAVA_HOME='$mainDir/java'" >> $HOME/.bashrc
echo "declare -x JAVA_OPTS='-server -Xmx1028m -XX:MaxPermSize=320m -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true'" >> $HOME/.bashrc
echo "declare -x MAVE2_HOME='/etc/maven2'">> $HOME/.bashrc
echo "declare -x MAVEN_OPTS='-Xms128m -Xmx796m -XX:PermSize=64m -XX:MaxPermSize=172m'">> $HOME/.bashrc
echo "declare -x SUBVERSION_HOME='/etc/subversion'" >> $HOME/.bashrc
echo "declare -x CATALINA_BASE='$mainDir/tomcat5'" >> $HOME/.bashrc
echo "declare -x CATALINA_HOME='$mainDir/tomcat5'" >> $HOME/.bashrc
echo "PATH='$PATH:$mainDir/java/bin:$mainDir/tomcat5/bin'" >> $HOME/.bashrc
```

Instalacion2.sh

```
#2- Preparación de los archivos
mainDir=$HOME"/Archivos"
mkdir $mainDir
tar xvfz Material.tar.gz
cd Files

#3- Instalación Java
tar xvfz jdk-7u6-linux-i586.tar.gz
mv jdk1.7.0_06 $mainDir/java

#4- Instalación Mysql-serverls
sudo apt-get install mysql-server-5.5
sudo chmod 0666 /etc/mysql/my.cnf
sudo mv my.cnf /etc/mysql
sudo chmod 0644 /etc/mysql/my.cnf
```

#5- Crear la base de datos Sakai

```
echo ""
echo "Inserta la contraseña de mysql"
echo ""
echo "create database Sakai default character set utf8; grant all privileges on Sakai.* to 'Sakai'@'localhost' identified by 'ironchef'; flush privileges; show databases;" |/usr/bin/mysql -uroot -p
```

#6- Instalación Maven2

```
sudo apt-get install maven2
```

#7- Instalación Subversion

```
sudo apt-get install subversion
```

#8- Instalación Tomcat

```
tar xvfz apache-tomcat-5.5.35.tar.gz
tar xvfz apache-tomcat-5.5.35-admin.tar.gz
tar xvfz apache-tomcat-5.5.35-deployer.tar.gz
mv apache-tomcat-5.5.35-deployer/* apache-tomcat-5.5.35
rm -r apache-tomcat-5.5.35-deployer
mv apache-tomcat-5.5.35 $mainDir"/tomcat5"
cp server.xml $mainDir/tomcat5/conf/server.xml

cp -r $mainDir/tomcat5/webapps $mainDir/tomcat5/webold
rm -rf $mainDir/tomcat5/webapps/*
rm -rf $mainDir/tomcat5/logs/*
echo "export JAVA_OPTS='-server -Xmx1028m -XX:MaxPermSize=320m -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true'" >> $mainDir/tomcat5/bin/setenv.sh
```

#9- Instalación connector MySQL

```
tar xvf mysql-connector-java-5.1.21.tar.gz
cp mysql-connector-java-5.1.21/mysql-connector-java-5.1.21-bin.jar $mainDir/tomcat5/common/lib/
rm -r mysql-connector-java-5.1.21
```

#10- Instalación Sakai

```
mkdir $mainDir/Sakai
cd $mainDir/Sakai
svn checkout https://source.Sakaiproject.org/svn/Sakai/branches/Sakai-2.8.x/ $mainDir/Sakai
```

#11- Configuración Sakai.properties

```
mkdir $mainDir/tomcat5/Sakai
cp Sakai.properties $mainDir/tomcat5/Sakai/Sakai.properties
```

#12- Crear settings.xml

```
echo "<settings xmlns=\"http://maven.apache.org/POM/4.0.0\"" > $HOME/.m2/settings.xml
echo "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"" >> $HOME/.m2/settings.xml
echo "xsi:schemaLocation=\"http://maven.apache.org/POM/4.0.0" >> $HOME/.m2/settings.xml
echo "http://maven.apache.org/xsd/settings-1.0.0.xsd\">" >> $HOME/.m2/settings.xml
echo "<profiles>" >> $HOME/.m2/settings.xml
echo "<profile>" >> $HOME/.m2/settings.xml
echo "<id>tomcat5x</id> " >> $HOME/.m2/settings.xml
echo "<activation>" >> $HOME/.m2/settings.xml
echo "<activeByDefault>true</activeByDefault>" >> $HOME/.m2/settings.xml
echo "</activation>" >> $HOME/.m2/settings.xml
echo "<properties>" >> $HOME/.m2/settings.xml
echo "<appserver.id>tomcat5x</appserver.id>" >> $HOME/.m2/settings.xml
echo "<appserver.home>$mainDir/tomcat5</appserver.home> " >> $HOME/.m2/settings.xml
echo "<maven.tomcat.home>$mainDir/tomcat5</maven.tomcat.home> " >>
$HOME/.m2/settings.xml
echo "<Sakai.appserver.home>$mainDir/tomcat5</Sakai.appserver.home> " >>
$HOME/.m2/settings.xml
echo "<surefire.reportFormat>plain</surefire.reportFormat>" >> $HOME/.m2/settings.xml
echo "<surefire.useFile>false</surefire.useFile>" >> $HOME/.m2/settings.xml
echo "</properties>" >> $HOME/.m2/settings.xml
echo "</profile>" >> $HOME/.m2/settings.xml
echo "</profiles>" >> $HOME/.m2/settings.xml
echo "</settings>" >> $HOME/.m2/settings.xml
```

#13- Construir Sakai usando Maven

```
cd $mainDir/Sakai
mvn -Pcafe clean install
mvn -Pcafe Sakai:deploy
```

#14-Iniciar Tomcat

```
cd $mainDir/tomcat5/bin/
sh startup.sh
```

#15- Instalación Eclipse

```
sudo apt-get install eclipse
```

Ahora pasaremos a explicar detalladamente el código del script:

Paso 1: Declararemos las variables de entorno

```
echo "declare -x JAVA_HOME='$mainDir/java'" >> $HOME/.bashrc
echo "declare -x JAVA_OPTS='-server -Xmx1028m -XX:MaxPermSize=320m -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true'" >> $HOME/.bashrc
echo "declare -x MAVE2_HOME='/etc/maven2'">> $HOME/.bashrc
echo "declare -x MAVEN_OPTS='-Xms128m -Xmx796m -XX:PermSize=64m -XX:MaxPermSize=172m'">> $HOME/.bashrc
echo "declare -x SUBVERSION_HOME='/etc/subversion'" >> $HOME/.bashrc
echo "declare -x CATALINA_BASE='$mainDir/tomcat5'" >> $HOME/.bashrc
echo "declare -x CATALINA_HOME='$mainDir/tomcat5'" >> $HOME/.bashrc
echo "PATH='$PATH:$mainDir/java/bin:$mainDir/tomcat5/bin'" >> $HOME/.bashrc
```

En este apartado definimos las variables de entorno que necesitan los diferentes componentes de Sakai para su correcto funcionamiento. Estas variables se definen en el archivo .bashrc localizado en la carpeta home de cada usuario, ya que este archivo se carga cada vez que se abre un terminal.

Paso 2: Preparación de los archivos

```
mainDir=$HOME"/Archivos"
mkdir $mainDir
tar xvfz Material.tar.gz
cd Files
```

En este apartado creamos la carpeta Archivos en el directorio home del usuario para no tener que usar los privilegios de root, ya que para poder acceder a /opt como se describe en la instalación se necesitan privilegios de súper usuario y esto a largo plazo da errores de permisos. Descomprimimos el archivo comprimido en el que se localizan todos los archivos que necesitamos para la instalación y que usa en el script.

Paso 3: Instalación Java

```
tar xvfz jdk-7u6-linux-i586.tar.gz
mv jdk1.7.0_06 $mainDir/java
```

En este apartado descomprimimos el archivo que contiene los archivos binarios para ejecutar java y movemos esta carpeta al directorio que hemos creado anteriormente que contendrá los archivos necesarios para Sakai.

Paso 4: Instalación Mysql-serverls

```
sudo apt-get install mysql-server-5.5
sudo chmod 0666 /etc/mysql/my.cnf
sudo mv my.cnf /etc/mysql
sudo chmod 0644 /etc/mysql/my.cnf
```

En este apartado instalamos MySQL server, el gestor de base de datos que usara Sakai. La instalaremos des de los repositorios de Linux que es la forma más sencilla y rápida de instalarla. Sustituiremos el archivo my.cnf que previamente nosotros hemos modificado para que funcione correctamente con Sakai.

Paso 5: Crear la base de datos Sakai

```
echo ""
echo "Inserta la contraseña de MySQL"
echo ""
echo "create database Sakai default character set utf8; grant all privileges on Sakai.* to 'Sakai'@'localhost' identified by 'ironchef'; flush privileges; show databases;" |/usr/bin/mysql -uroot -p
```

En este apartado creamos la base de datos llamada Sakai con el usuario Sakai y la contraseña ironchef. Esta es la base de datos que usara Sakai.

Paso 6: Instalación Maven2

```
sudo apt-get install maven2
```

En este apartado instalaremos Maven2 que se encargara de gestionar y compilar los proyectos de Sakai. Lo instalaremos desde repositorio ya que es la forma más fácil y rápida de instalar programas en Linux.

Paso 7: Instalación Subversion

```
sudo apt-get install subversion
```

En este apartado instalaremos subversión que se encargara de la gestión de las versiones de Sakai. Lo instalaremos desde repositorio ya que es la forma más fácil y rápida de instalar programas en Linux.

Paso 8: Instalación Tomcat

```
tar xvfz apache-tomcat-5.5.35.tar.gz
tar xvfz apache-tomcat-5.5.35-admin.tar.gz
tar xvfz apache-tomcat-5.5.35-deployer.tar.gz
mv apache-tomcat-5.5.35-deployer/* apache-tomcat-5.5.35
rm -r apache-tomcat-5.5.35-deployer
mv apache-tomcat-5.5.35 $mainDir"/tomcat5"
cp server.xml $mainDir/tomcat5/conf/server.xml

cp -r $mainDir/tomcat5/webapps $mainDir/tomcat5/webold
rm -rf $mainDir/tomcat5/webapps/*
rm -rf $mainDir/tomcat5/logs/*
echo "export JAVA_OPTS='-server -Xmx1028m -XX:MaxPermSize=320m -Dorg.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING=false -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dsun.lang.ClassLoader.allowArraySyntax=true'" >> $mainDir/tomcat5/bin/setenv.sh
```

En este apartado vamos a instalar el contenedor de Servlet sobre el que funciona Sakai. Para instalar vamos a descomprimir el Tomcat core, el Tomcat admin y el Tomcat deployer todo en el mismo directorio que constituyen la estructura necesaria para el funcionamiento de Sakai. Este directorio lo movemos al mismo directorio que reside el directorio por motivos de configuración y de orden. Substituimos el archivo server.xml que hemos modificado para que utilice la codificación UTF-8. Borramos el contenido de la carpeta webapps y el de la carpeta logs, esto es opcional pero lo hacemos para tener solo lo relacionado con Sakai. Y por último creamos el archivo setenv.sh en el directorio de archivos binarios con la variable de entorno JAVA_OPTS.

Paso 9: Instalación connector MySQL

```
tar xvf mysql-connector-java-5.1.21.tar.gz
cp mysql-connector-java-5.1.21/mysql-connector-java-5.1.21-bin.jar $mainDir/tomcat5/common/lib/
rm -r mysql-connector-java-5.1.21
```

En este apartado instalaremos el conector java MySQL, que lo que hace es conectar java con la el gestor de base de datos MySQL. Para ello vamos a descomprimir el archivo mysql-connector-java-5.1.21.tar.gz y copiar el conector en el directorio base de tomcat/common/lib.

Paso 10: Instalación Sakai

```
mkdir $mainDir/Sakai
cd $mainDir/Sakai
svn checkout https://source.Sakaiproject.org/svn/Sakai/branches/Sakai-2.8.x/ $mainDir/Sakai
```

En este apartado nos descargamos mediante subversión los archivos binarios de Sakai en el directorio de Archivos que hemos creado al principio. Estos archivos se compilaran posteriormente y se copiaran en el directorio webapps situada en el directorio base de Tomcat.

Paso 11: Configuración Sakai.properties

```
mkdir $mainDir/tomcat5/Sakai
cp Sakai.properties $mainDir/tomcat5/Sakai/Sakai.properties
```

En este apartado copiamos archivo Sakai.properties el directorio Sakai situado en la carpeta base de Tomcat. Este archivo lo hemos modificado y configurado para la base de datos que hemos creado en el paso 5.

Paso 12: Crear settings.xml

```
echo "<settings xmlns=\"http://maven.apache.org/POM/4.0.0\"" > $HOME/.m2/settings.xml
echo "xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"" >> $HOME/.m2/settings.xml
echo "xsi:schemaLocation=\"http://maven.apache.org/POM/4.0.0" >> $HOME/.m2/settings.xml
echo "http://maven.apache.org/xsd/settings-1.0.0.xsd\">" >> $HOME/.m2/settings.xml
echo "<profiles>" >> $HOME/.m2/settings.xml
echo "<profile>" >> $HOME/.m2/settings.xml
echo "<id>tomcat5x</id> " >> $HOME/.m2/settings.xml
echo "<activation>" >> $HOME/.m2/settings.xml
echo "<activeByDefault>true</activeByDefault>" >> $HOME/.m2/settings.xml
echo "</activation>" >> $HOME/.m2/settings.xml
echo "<properties>" >> $HOME/.m2/settings.xml
echo "<appserver.id>tomcat5x</appserver.id>" >> $HOME/.m2/settings.xml
echo "<appserver.home>$mainDir/tomcat5</appserver.home> " >> $HOME/.m2/settings.xml
echo "<maven.tomcat.home>$mainDir/tomcat5</maven.tomcat.home> " >>
$HOME/.m2/settings.xml
echo "<Sakai.appserver.home>$mainDir/tomcat5</Sakai.appserver.home> " >>
$HOME/.m2/settings.xml
echo "<surefire.reportFormat>plain</surefire.reportFormat>" >> $HOME/.m2/settings.xml
echo "<surefire.useFile>false</surefire.useFile>" >> $HOME/.m2/settings.xml
echo "</properties>" >> $HOME/.m2/settings.xml
echo "</profile>" >> $HOME/.m2/settings.xml
echo "</profiles>" >> $HOME/.m2/settings.xml
echo "</settings>" >> $HOME/.m2/settings.xml
```

En este apartado creamos dinámicamente el archivo settings.xml porque en este reside la ruta del directorio base Tomcat que se utiliza Maven durante la compilación, instalación y despliegue de Sakai y de los proyectos que se desarrollan para Sakai. Es creado dinámicamente porque la ruta de este directorio es diferente en cada máquina y en cada usuario

Paso 13: Construir Sakai usando Maven

```
cd $mainDir/Sakai
mvn -Pcafe clean install
mvn -Pcafe Sakai:deploy
```

En este apartado nos situamos en el directorio que nos hemos bajado los archivos binarios de Sakai y ejecutamos Maven. Lo que hace Maven es compilar y copiar los archivos compilados (Servlets) al contenedor de Servlets, en nuestro caso al directorio webapps de apache Tomcat.

Paso 14: Iniciar Tomcat

```
cd $mainDir/tomcat5/bin/
sh startup.sh
```

En este apartado iniciamos el Tomcat, para ello nos situamos en el directorio donde se localizan los archivos binarios de apache Tomcat y ejecutamos el archivo startup.sh que inicia el Tomcat.

Paso 15: Instalación Eclipse

```
sudo apt-get install eclipse
```

En este apartado instalamos el Eclipse, no es necesario para Sakai. Eclipse se utiliza para desarrollar aplicaciones para Sakai. Para su instalación utilizamos los repositorios de Linux que es la forma más sencilla y fácil de instalarlo. Para poder desarrollar aplicaciones de Sakai en Linux debemos configurar adecuadamente Eclipse. La configuración se debe realizar desde Eclipse ya que no se puede hacer desde el terminal.

CAPITULO 5: PRIMERA APLICACIÓN DE SAKAI

Para desarrollar aplicaciones necesitamos instalar un entorno de desarrollo y configurarlo para que funcionen correctamente en Sakai.

Paso 1: Instalación de Eclipse

a. Para instalar Eclipse debemos descargarnos la última versión estable que encontremos en el siguiente enlace:

http://www.eclipse.org/

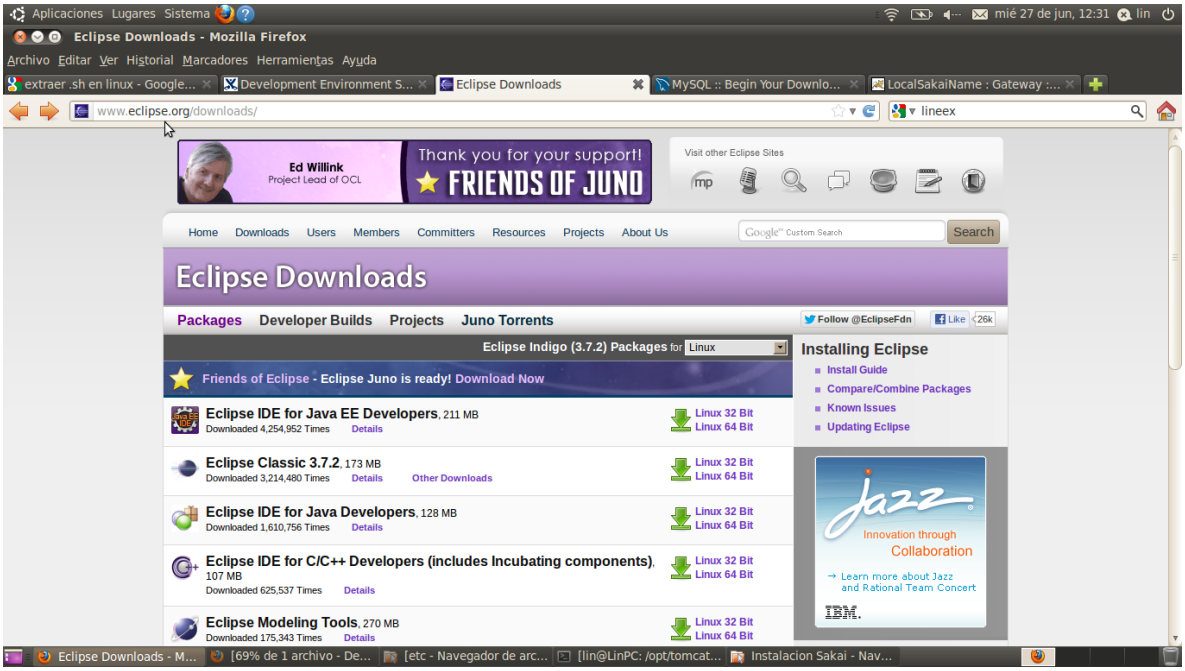

Figura 33: Descargar Eclipse

b. Extraemos el archivo descargado en el directorio /opt

c. Ejecutamos Eclipse para verificar que trabaja adecuadamente.

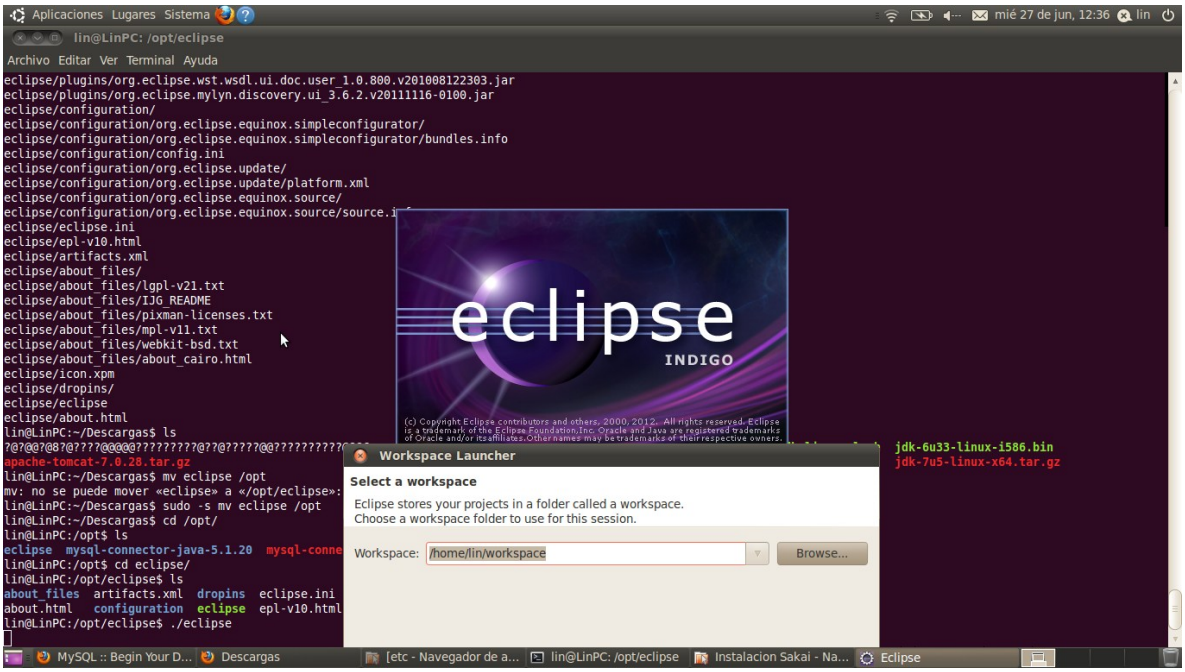

Figura 34: Ejecución Eclipse

d. Establecemos los ajustes de memoria para eclipse:

Apagamos eclipse si aún estaba ejecutándose. Editamos el archivo eclipse.ini que se encuentra en el directorio /opt/eclipse, que es desde donde arrancamos eclipse, y cambiamos los siguientes ajustes:

-vmargs
-Xms40m
-Xmx256m

Por estos:
--launcher.XXMaxPermSize
256M
-vmargs
-Xms128m
-Xmx1024m
-XX:+UseParallelGC

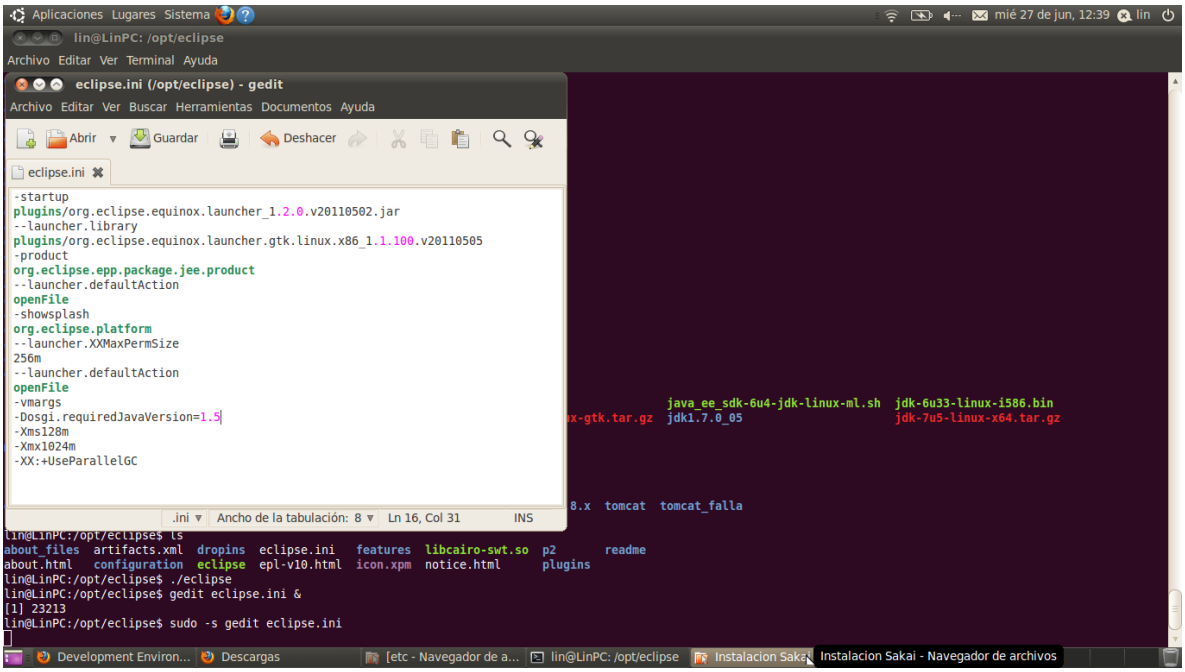

Figura 35: Ajustes memoria Eclipse

e. Establecemos el JVM

Para establecer el JVM ejecutamos Eclipse si no está ya en ejecución. Nos dirigimos al apartado Windows → Preferences → Java → Installed JREs y seleccionamos la versión actual del JRE instalado. Clicamos el botón de Edit, luego Browse y navegamos al directorio en el que tenemos instalado el JAVA. Le damos a OK y finalizamos el proceso.

Paso 2: Agregar Subeclipse a Eclipse

a. Iniciamos Eclipse

b. Vamos a Help → Software Updates

c. Seleccionamos Avalaible Software y clicamos en continuar.

d. Agregamos el siguiente sitio con el Botón Add Site:

http://subclipse.tigris.org/update_1.6.x

Y finalizamos.

e. Seleccionamos la opción de Subclipse, le damos a continuar, aceptamos los términos y condiciones y continuamos dándole al botón next hasta que esté todo instalado.

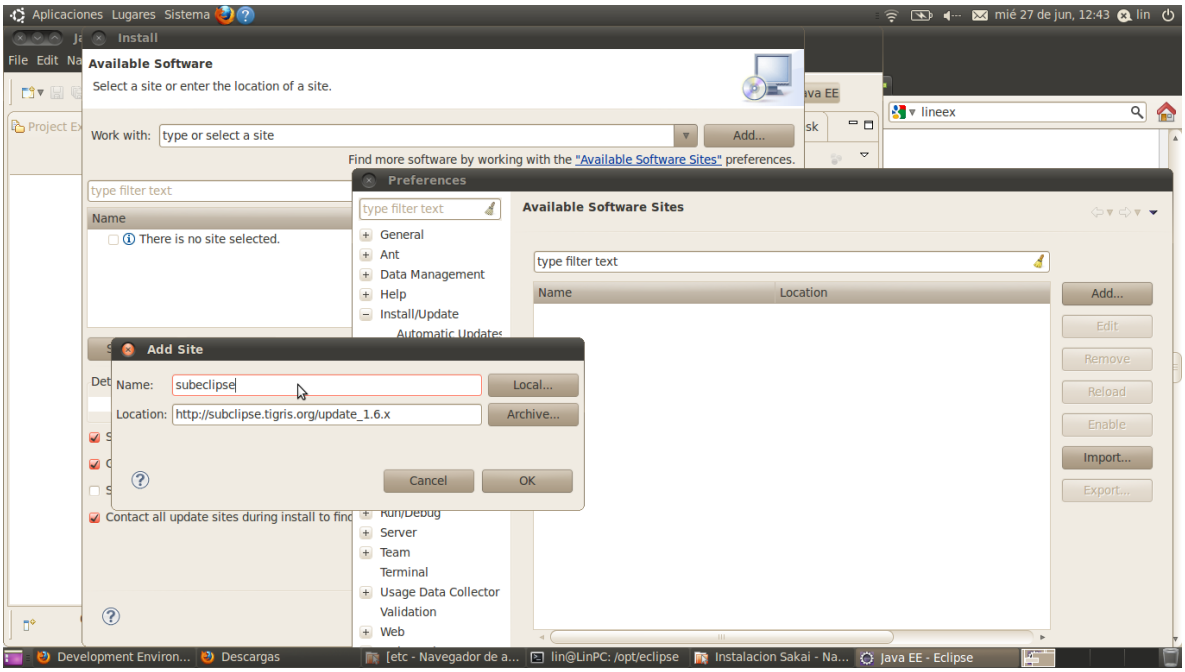

Figura 36: Agregación Subeclipse

f. Reiniciamos el programa cuando pregunte.

g. Si queremos evitar que Subversion actualice los archivos “bin”, “target” y “m2-target” debemos ir a Windows → Preferences y seleccionar Team → Ignored Resources, agregamos los “bin”, “target” y “m2-target” con Add Patern y le damos a aceptar.

Paso 3: Agregar el plugin de Maven a Eclipse

a. Iniciamos Eclipse
b. Vamos a Help → Software Updates, seleccionamos Avalaible Software y le damos a continuar.
c. Agregamos el siguiente enlace con Add Site:
http://m2eclipse.sonatype.org/sites/m2e
Y hacemos clic finalizar.

d. Seleccionamos Maven Integration y hacemos clic en continuar. Aceptamos los términos y condiciones y hacemos clic en continuar hasta que la instalación este completa.
Finalmente reiniciamos el programa cuando pregunte.

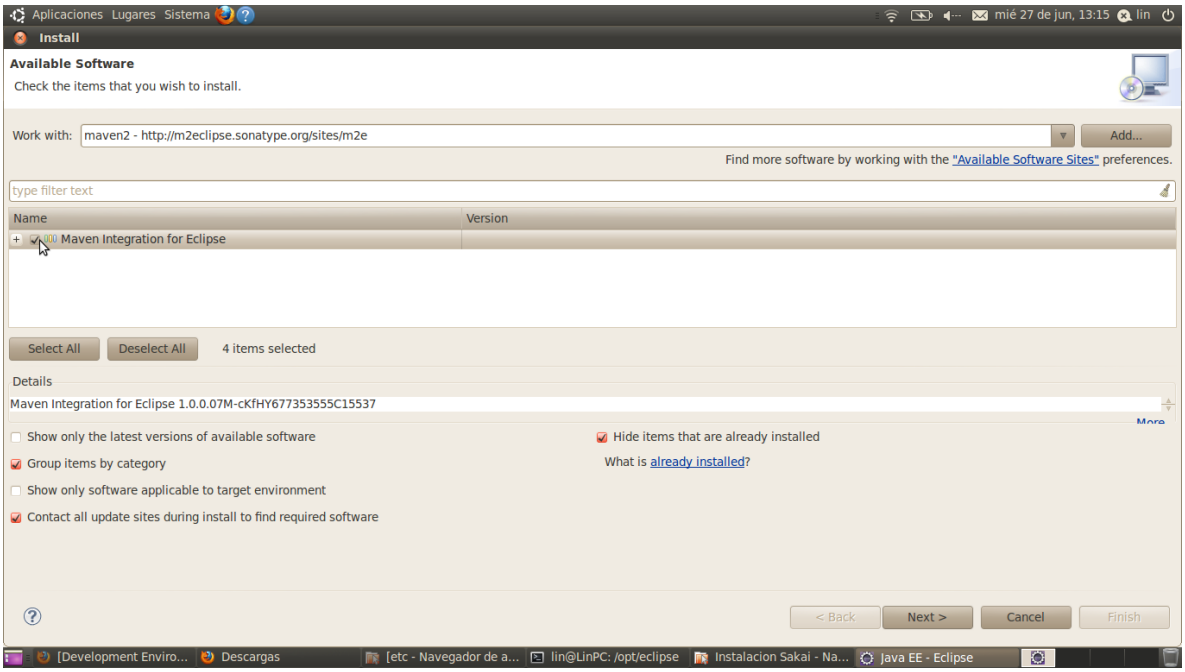

Figura 37: Agregación de maven a Eclipse

4. Crear proyecto Sakai

El último punto del tutorial no nos funciona por eso hacemos el proyecto de la aplicación de Sakai mediante wicket. Para hacer el proyecto wicket se copia la siguiente línea en el terminal de Linux:

$ mvn archetype:generate -DarchetypeGroupId=org.Sakaiproject.maven-archetype -DarchetypeArtifactId=Sakai-wicket-maven-archetype \ -DarchetypeVersion=1.2 -DarchetypeRepository=https://source.Sakaiproject.org/maven2/

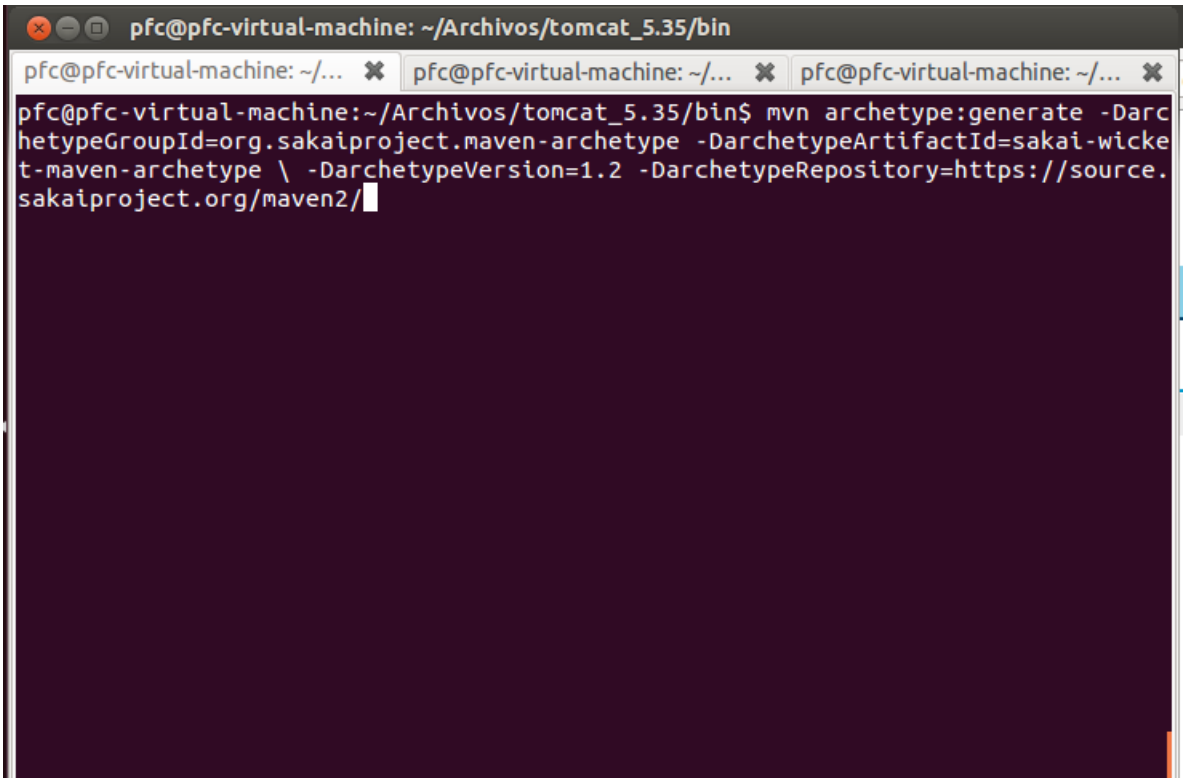

Figura 38: Creación proyecto Sakai

Después de esto nos preguntara tres cosas que son:

GroupId: Aquí se escribe el nombre del paquete del proyecto.

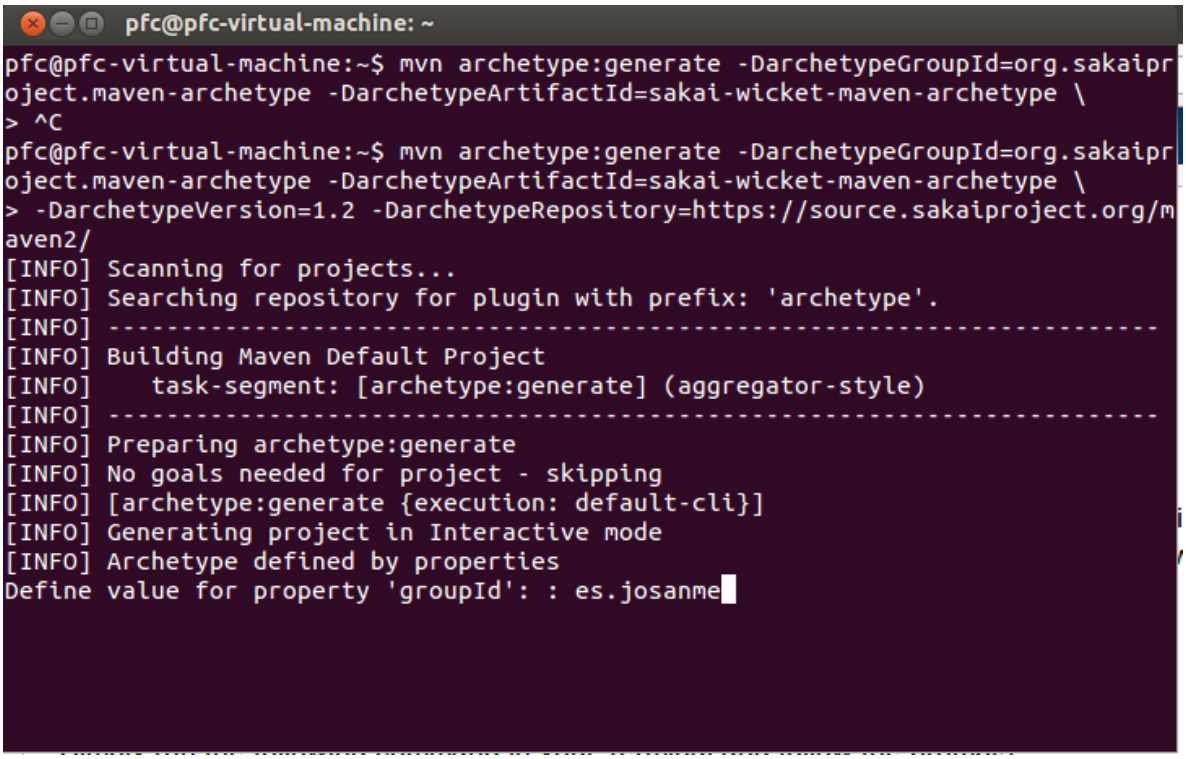

Figura 39: Valor GroupId

ArtifactId: Aquí se escribe el nombre del proyecto.

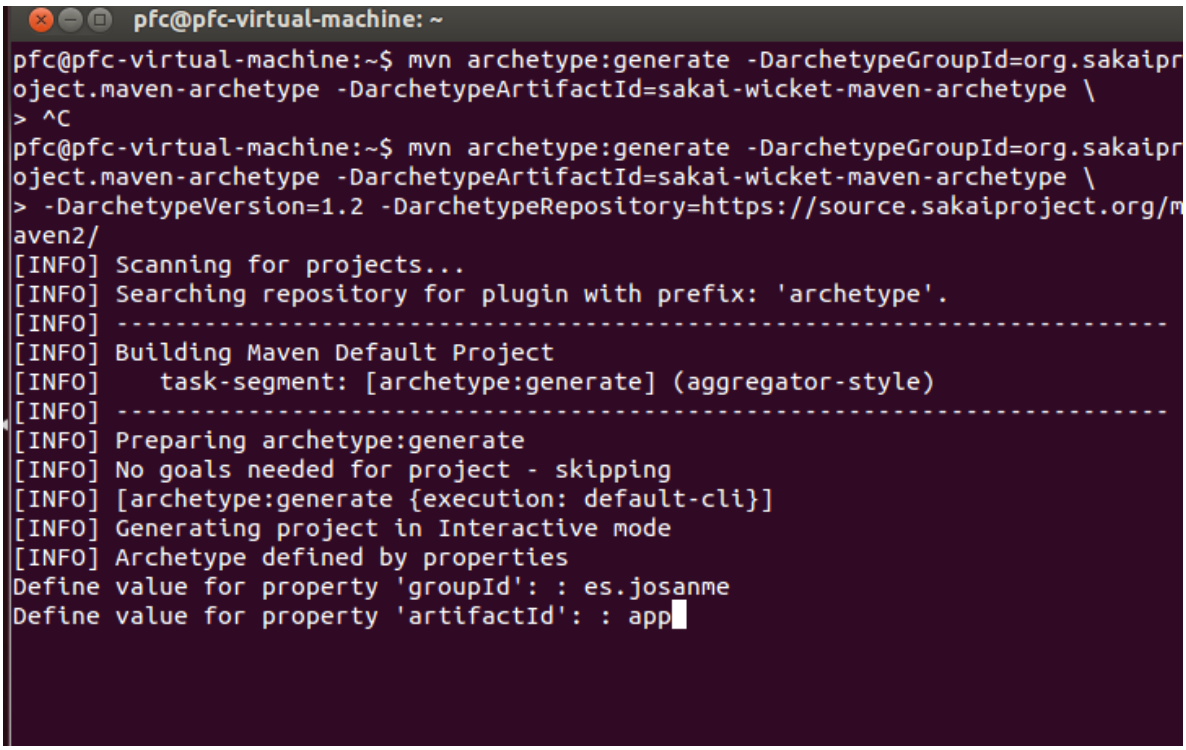

Figura 40: Nombre del proyecto

i la versión: que la dejaremos por defecto (snapshot).

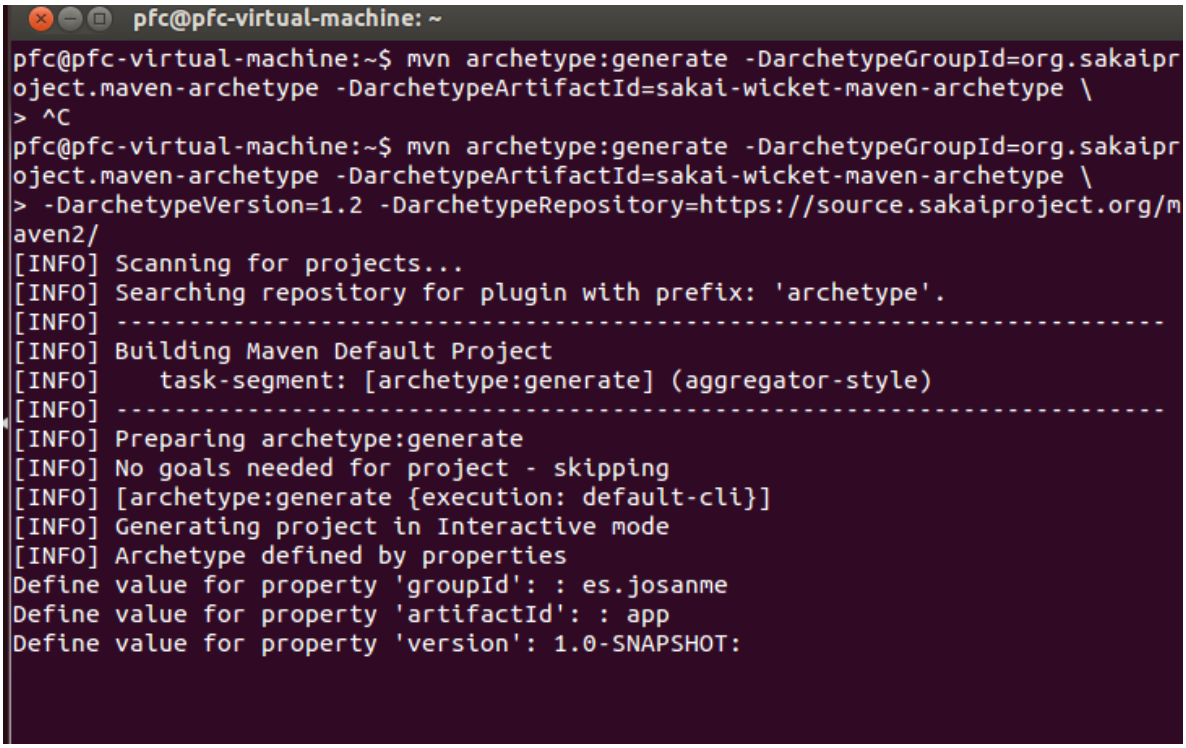

Figura 41: versión del proyecto

Después de todo esto tenemos un proyecto hola mundo se crea una carpeta con el nombre del proyecto que hemos escrito anteriormente. Ahora nos metemos dentro de la nueva carpeta con el comando:

$ cd app

Una vez dentro compilaremos e instalaremos el proyecto dentro de Sakai mediante la orden:

$ mvn clean install Sakai:deploy

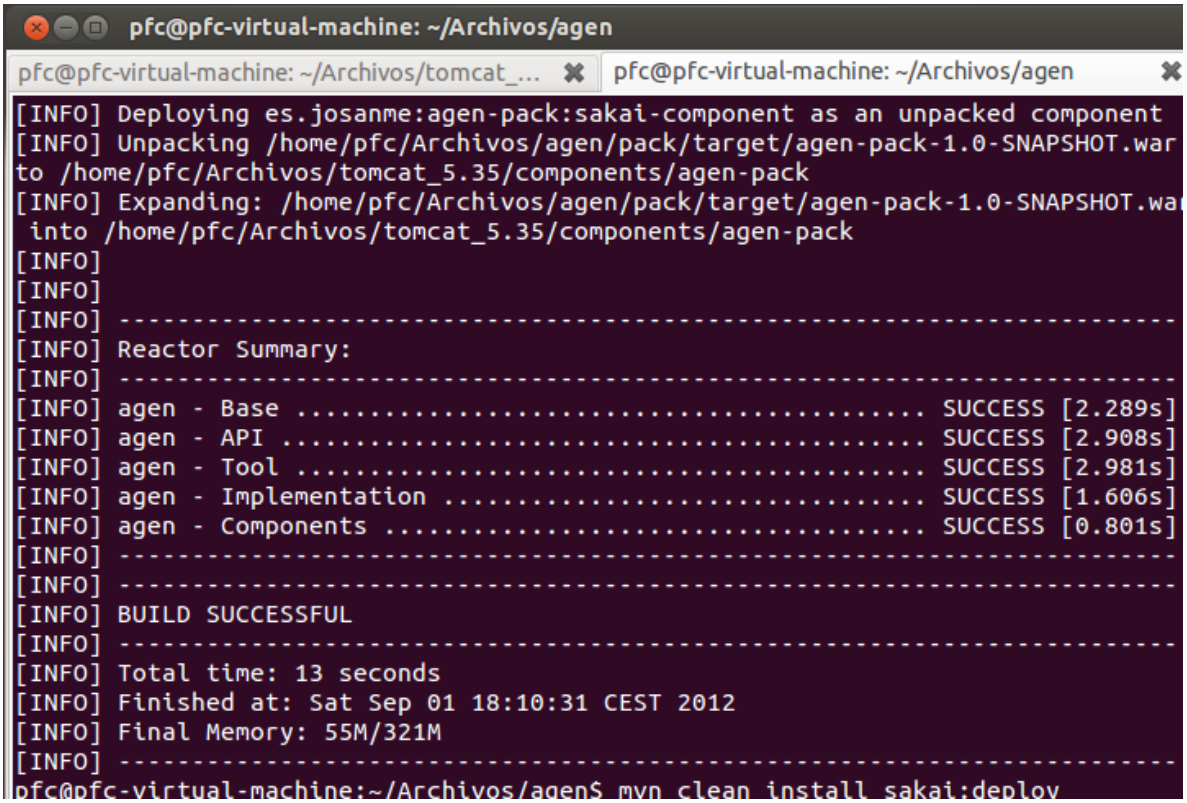

Figura 42: Construcción Completa del proyecto wicket

Después de esto ya tenemos la aplicación lista para ejecutarse desde Sakai. Ahora para verlo tenemos que crear un nuevo sitio dentro de Sakai. Para ello vamos a configuración de sitios y pulsamos nuevo.

Después seleccionamos new Project page y pulsamos continuar.

Ahora escribimos el nombre del nuevo sitio y pulsamos continuar.

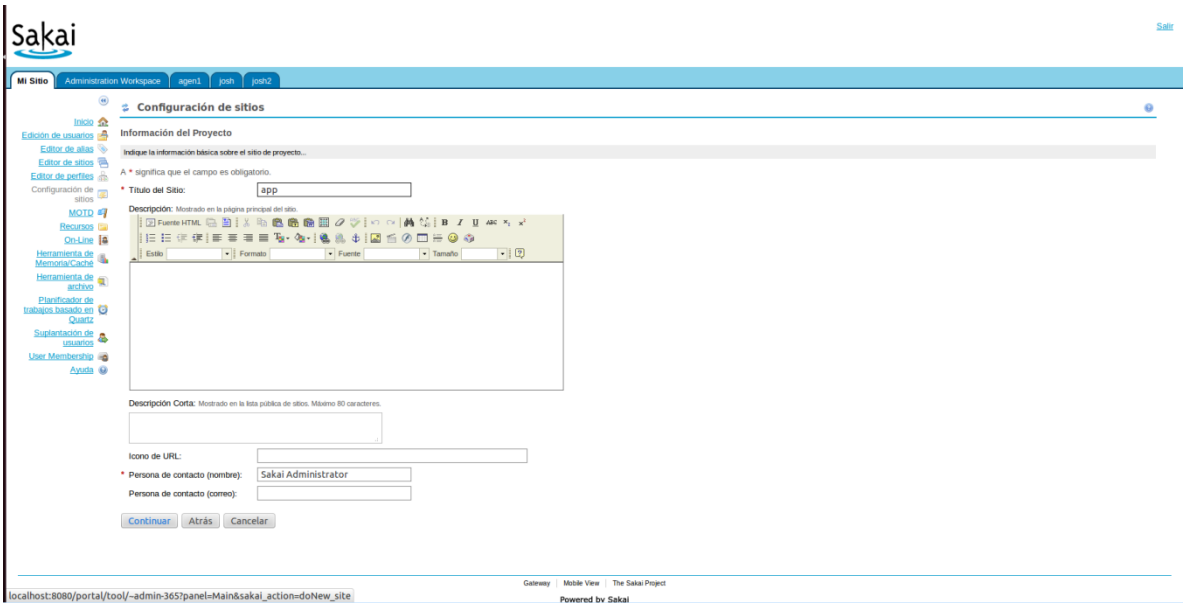

Figura 43: Sitio en Sakai

Seleccionamos My Sakay app y pulsamos continuar.

Pulsamos Continuar y ya tenemos nuestro nuevo sitio con la nueva aplicación. Ahora tendremos una nueva pestaña arriba con el nombre que hemos puesto cuando hemos creado el nuevo sitio.

Pulsamos sobre la pestaña en este caso sobre la pestaña “app” y después sobre My Sakai app y tenemos la aplicación que se crea por defecto un “Hello Sakai Administrator”

Figura 44: Aplicación en Sakai

En este momento ya tenemos la primera aplicación de Sakai funcionando. En el siguiente Capitulo Modificaremos este proyecto y crearemos nuestra aplicación.

CAPITULO 6: DESARROLLO DE APLICACIÓN DE UNA AGENDA DE CONTACTOS

La aplicación que hemos desarrollado consiste en una sencilla agenda. La aplicación está formada por 7 clases java, que son Contacto, MySQL, MyAplication, BasePage, FirstPage, SecondPager y ViewContact. También consta de 4 archivos html que son necesarios en wicket, en estos se indica donde se sitúan los diferente componentes que se modifican desde java. A continuación vamos a explicar más detalladamente el código de la aplicación.

```
public class Contacto  implements Serializable{
	private String name;
	private String surName;
	private String email;
	private String phone;
	public Contacto(String name,String surname, String email ,String phone)
	{
		this.name = name;
		this.surName = surname;
		this.email = email;
		this.phone = phone;

	}

	public String getName(){return name;}
	public String getPhone(){return phone;}
	public String getSurName(){return surName;}
	public String getEmail(){return email;}

	public void setName(String name){this.name = name;}
	public void setPhone(String phone){this.phone = phone;}
	public void setSurName(String surName){this.surName = surName;}
	public void setEmail(String email){this.email = email;}
}
```

Listado 4. Clase Contacto

Esta es la clase Contacto, es la que contiene toda la información necesaria para definir el contacto. Contiene el nombre, apellido, email y teléfono, esta es la estructura que definido yo como un contacto. Está formada por un constructor, 4 consultores y 4 modificadores. Los consultores devuelve el dato del atributo correspondiente con dicho consultor y los modificadores modifican el valor del dato correspondiente. La modificadora sobra porque no los uso, los tengo para futuras versiones en la que se pueda editar un contacto.

```
package es.josanme.model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class Mysql {
	Connection conn = null;
	Statement stmt = null;
	ResultSet rs = null;
	public void con() {
		try {
			conn = DriverManager
	.getConnection("jdbc:mysql://localhost/Sakai?user=Sakai&password=ironchef");
			stmt = conn.createStatement();
			stmt.executeUpdate("CREATE TABLE IF NOT EXISTS agenda (id int not null
auto_increment,primary key(id),name varchar(20),surname varchar(50),email varchar(50),tlf
varchar(15))");
		} catch (SQLException ex) {
		}
	}
	public void close() {
		try {
			if (conn != null)
				conn.close();
			if (rs != null)
				rs.close();
			if (stmt != null)
				stmt.close();
		} catch (SQLException sqlEx) {
		}
	}
	public ResultSet list() {
		con();
		try {
			rs = stmt.executeQuery("select name,surname,email,tlf from agenda order
by name asc");
		} catch (SQLException e) {
		}
		return rs;
	}
```

```
	public boolean newContact(Contacto cont) {
		int numRow = -1;
		boolean state = false;
		con();
		try {
			rs = stmt.executeQuery("select count(*) from agenda where
name='"+cont.getName()+"' and email= '"+cont.getEmail()+"'");
			rs.next();
			if(rs.getInt(1)==0)
			{
				numRow = stmt.executeUpdate("insert into agenda
(name,surname,email,tlf) values('"
				+ cont.getName() + "','" + cont.getSurName() +
"','"+cont.getEmail()+"','"+cont.getPhone()+"')");
				stmt.executeUpdate("commit");
			}
		} catch (SQLException e) {}
		close();
		if (numRow > 0)
			state = true;
		return state;

	}

	public boolean deleteContact(Contacto cont) {
		int numRow = -1;
		boolean state = false;
		int id;
		con();
		try {
			rs = stmt.executeQuery("select id from agenda where
name='"+cont.getName()+"' and email = '"+cont.getEmail()+"'");
			rs.next();
			id = rs.getInt(1);
			numRow = stmt.executeUpdate("delete from agenda where id ="+ id );
			stmt.executeUpdate("commit");
		} catch (SQLException e) {
		}
		close();
		if (numRow > 0)
			state = true;
		return state;

	}

}
```

Listado 5. Clase MySQL

Esta es la clase Mysql es la encargada de la comunicación con la base de datos de Sakai. Está formada por 5 métodos que son:

- Con(), este se encarga de conectar con la base de datos y crear la tabla de mysql que necesita la aplicación para su funcionamiento.
- Close(), se encarga de cerrar todo los objetos que se han abierto durante la conexión.
- List(), este devuelve un objeto que contiene todos los contacto de la base de datos ordenado s por nombre ascendentemente.
- newContact(), a este se le pasa el contacto, mira en la base de datos si existe ese contacto con ese nombre y ese email, si existe devuelve falso y si no existe inserta el contacto a la base de datos y devuelve verdadero.
- deleteContact(), a este se le pasa un contacto, lo busca y lo elimina de la base de datos.

```
package es.josanme.tool;

import org.apache.wicket.Page;
import org.apache.wicket.Request;
import org.apache.wicket.RequestCycle;
import org.apache.wicket.Response;
import org.apache.wicket.protocol.http.WebApplication;
import org.apache.wicket.protocol.http.WebRequest;
import org.apache.wicket.protocol.http.WebRequestCycle;
import org.apache.wicket.protocol.http.WebResponse;
import org.apache.wicket.spring.injection.annot.SpringComponentInjector;
import es.josanme.tool.pages.FirstPage;

public class MyApplication extends WebApplication {

	@Override
	protected void init() {

		// Configure for Spring injection
		addComponentInstantiationListener(new SpringComponentInjector(this));

		// Don't throw an exception if we are missing a property, just fallback
		getResourceSettings().setThrowExceptionOnMissingResource(false);

		// Remove the wicket specific tags from the generated markup
		getMarkupSettings().setStripWicketTags(true);

		// Don't add any extra tags around a disabled link (default is
		// <em></em>)
		getMarkupSettings().setDefaultBeforeDisabledLink(null);
		getMarkupSettings().setDefaultAfterDisabledLink(null);

		// On Wicket session timeout, redirect to main page
```

```
                getApplicationSettings().setPageExpiredErrorPage(FirstPage.class);
                getApplicationSettings().setAccessDeniedPage(FirstPage.class);

                // to put this app into deployment mode, see web.xml

        }

        @Override
        public RequestCycle newRequestCycle(Request request, Response response) {
                return new WebRequestCycle(this, (WebRequest) request,
                                (WebResponse) response) {
                        @Override
                        public Page onRuntimeException(Page page, RuntimeException e) {
                                throw e;
                        }
                };
        }
        public Class getHomePage() {
                return FirstPage.class;
        }

        public MyApplication() {
        }

}
```

Listado 6. Clase MyAppilcation

Esta es la clase MyApplication, es la que se ejecuta nada más lanzar la aplicación de Sakai. Esta hereda de la clase WebApplication que es la clase necesaria y mínima para ejecutar aplicaciones web. Esta clase sobrescribe el método init() que configura la aplicación como deseemos. Consta también de un método getHomePage() que es el que lanza la página que se ve nada más abrir la aplicación.

```
package es.josanme.tool.pages;

import org.apache.log4j.Logger;
import org.apache.wicket.AttributeModifier;
import org.apache.wicket.Component;
import org.apache.wicket.behavior.AttributeAppender;
import org.apache.wicket.behavior.SimpleAttributeModifier;
import org.apache.wicket.feedback.FeedbackMessage;
import org.apache.wicket.markup.html.IHeaderContributor;
import org.apache.wicket.markup.html.IHeaderResponse;
import org.apache.wicket.markup.html.WebPage;
```

```
import org.apache.wicket.markup.html.basic.Label;
import org.apache.wicket.markup.html.link.Link;
import org.apache.wicket.markup.html.panel.FeedbackPanel;
import org.apache.wicket.model.Model;
import org.apache.wicket.model.ResourceModel;
import org.apache.wicket.model.StringResourceModel;
import org.apache.wicket.spring.injection.annot.SpringBean;
import es.josanme.logic.SakaiProxy;

public class BasePage extends WebPage implements IHeaderContributor {
	private static final Logger log = Logger.getLogger(BasePage.class);
	@SpringBean(name="es.josanme.logic.SakaiProxy")
	protected SakaiProxy SakaiProxy;
	Link<Void> firstLink;
	Link<Void> secondLink;
	FeedbackPanel feedbackPanel;

	public BasePage() {
		log.debug("BasePage()");

		//first link
		firstLink = new Link<Void>("firstLink") {
			private static final long serialVersionUID = 1L;
			public void onClick() {
				setResponsePage(new FirstPage());
			}
		};
		firstLink.add(new Label("firstLinkLabel",new
ResourceModel("link.first")).setRenderBodyOnly(true));
		firstLink.add(new AttributeModifier("title", true, new
ResourceModel("link.first.tooltip")));
		add(firstLink);

		//second link
		secondLink = new Link<Void>("secondLink") {
			private static final long serialVersionUID = 1L;
			public void onClick() {
				setResponsePage(new SecondPage());
			}
		};
		secondLink.add(new Label("secondLinkLabel",new
ResourceModel("link.second")).setRenderBodyOnly(true));
		secondLink.add(new AttributeModifier("title", true, new
ResourceModel("link.second.tooltip")));
		add(secondLink);


		// Add a FeedbackPanel for displaying our messages
```

```
    feedbackPanel = new FeedbackPanel("feedback"){

        @Override
        protected Component newMessageDisplayComponent(final String id, final
FeedbackMessage message) {
                final Component newMessageDisplayComponent =
super.newMessageDisplayComponent(id, message);

                if(message.getLevel() == FeedbackMessage.ERROR ||
                        message.getLevel() == FeedbackMessage.DEBUG ||
                        message.getLevel() == FeedbackMessage.FATAL ||
                        message.getLevel() == FeedbackMessage.WARNING){
                        add(new SimpleAttributeModifier("class", "alertMessage"));
                } else if(message.getLevel() == FeedbackMessage.INFO){
                        add(new SimpleAttributeModifier("class", "success"));

                }
                return newMessageDisplayComponent;
        }
    };
    add(feedbackPanel);
  }
        public void clearFeedback(FeedbackPanel f) {
                if(!f.hasFeedbackMessage()) {
                        f.add(new SimpleAttributeModifier("class", ""));
                }
        }
        public void renderHead(IHeaderResponse response) {


                //get Sakai skin
                String skinRepo = SakaiProxy.getSkinRepoProperty();
                String toolCSS = SakaiProxy.getToolSkinCSS(skinRepo);
                String toolBaseCSS = skinRepo + "/tool_base.css";

                //Sakai additions
                response.renderJavascriptReference("/library/js/headscripts.js");
                response.renderCSSReference(toolBaseCSS);
                response.renderCSSReference(toolCSS);
                response.renderOnLoadJavascript("setMainFrameHeight( window.name )");

                //Tool additions (at end so we can override if required)
                response.renderString("<meta  http-equiv=\"Content-Type\"  content=\"text/html;
charset=UTF-8\" />");
                //response.renderCSSReference("css/my_tool_styles.css");
                //response.renderJavascriptReference("js/my_tool_javascript.js");
        }
```

```
        protected void disableLink(Link<Void> l) {
                l.add(new AttributeAppender("class", new Model<String>("current"), " "));
                l.setRenderBodyOnly(true);
                l.setEnabled(false);
        }
}
```

Listado 7. Clase BasePage

Esta clase es la BasePage, se encarga de establecer el estilo de Sakai y construye las dos pestañas que utilizamos para ver la lista de contactos y para crear un contacto. Esta clase es la que heredan el res de clases relacionadas con las páginas.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html
        xmlns:wicket="http://wicket.apache.org/dtds.data/wicket-xhtml1.4-strict.dtd">
<head>
<title><wicket:message key="page.title" /></title>
<style type="text/css">
body {
        background-color: #d0e4fe;
}
.agendacontent {
        padding: 20px;
        font-family: "Calibri";
        font-size: large;
}
.agendacontent a:link {
        text-decoration: none;
        color: white;
        font-size: large;
}
.agendacontent a:visited {
        text-decoration: none;
        color: white;
        font-size: large;
}
.agendacontent a:hover {
        text-decoration: none;
        font-weight: bold;
        color: #000099;
        font-size: large;
}
.agendacontent a:active {
        text-decoration: none;
        color: white;
```

```
        font-size: large;
}
#fnav {
        padding-top: 30px;
}
.agendacontent p {
        font-weight: bold;
        margin-left: 5px;
}
.agendacontent h1 {
        color: white;
        font-weight: bold;
        margin-bottom: 20px;;
}
.agendacontent h3 {
        color: white;
        font-weight: bold;
        margin-bottom: 20px;
}
</style>
</head>
<body>
        <div align="center" class="portletBody">
                <ul class="navIntraTool actionToolbar" role="menu">
                        <li class="firstToolBarItem" role="menuitem"><span> <a
                                        wicket:id="firstLink">                                  <span
wicket:id="firstLinkLabel">[firstLinkLabel]</span>
                                </a>
                        </span></li>
                        <li role="menuitem"><span> <a wicket:id="secondLink">
                                                <span
wicket:id="secondLinkLabel">[secondLinkLabel]</span>
                                </a>
                        </span></li>
                </ul>

                <span wicket:id="feedback">feedbackmessages will be put here</span>

                <wicket:child />

        </div>
</body>
</html>
```

Listado 8. HTML de la clase BasePage

Este es el archivo BasePage.html que relaciona las posiciones de la página web con los objetos que utiliza y modifica la clase BasePage. Al heredar el resto de páginas esta página, he definido un

pequeño css para hacer un poco más bonita la aplicación.

```
package es.josanme.tool.pages;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import org.apache.wicket.ResourceReference;
import org.apache.wicket.markup.html.basic.Label;
import org.apache.wicket.markup.html.form.Form;
import org.apache.wicket.markup.html.form.ImageButton;
import org.apache.wicket.markup.html.link.Link;
import org.apache.wicket.markup.html.navigation.paging.PagingNavigator;
import org.apache.wicket.markup.repeater.Item;
import org.apache.wicket.markup.repeater.data.DataView;
import org.apache.wicket.markup.repeater.data.ListDataProvider;
import org.apache.wicket.model.CompoundPropertyModel;
import org.apache.wicket.model.PropertyModel;
import es.josanme.model.Contacto;
import es.josanme.model.Mysql;

public class FirstPage extends BasePage {

	public FirstPage() {
		Contacto contact = null;
		ArrayList list = new ArrayList();

		final Mysql bd = new Mysql();
		ResultSet rs = bd.list();
		try{
		while (rs.next()) {
				contact = new
Contacto(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4));
				list.add(contact);
			}

			bd.close();
		} catch (SQLException e) {}


		 Form<FirstPage> addDeleteForm = new Form<FirstPage>("deleteForm",
		  new CompoundPropertyModel<FirstPage>(this));
		 add(addDeleteForm);
		 final DataView dataView = new DataView("simple", new ListDataProvider(
				list)) {
			public void populateItem(final Item item) {
```

```
                        Link view = new Link("id", item.getModel()) {
                    public void onClick() {
                        Contacto c = (Contacto) getModelObject();
                        setResponsePage(new ViewContact(c));
                    }
                };
                Contacto c = (Contacto)item.getModelObject();
                view.add(new Label("name",c.getName()));
                item.add(view);

                        }
                    };


                    dataView.setItemsPerPage(25);
                    addDeleteForm.add(dataView);
                    add(new PagingNavigator("navigator", dataView));
            }
}
```

Listado 9. Clase FirstPage

Esta es la clase FirstPage, es la clase que muestra la lista de contactos, para esto lo que hace es llamar al método list de la clase Mysql (vista anteriormente) y meter todos los contactos en una lista de contactos y pasárselo a un DataView de tipo enlace, esto lo hemos hecho para cuando hagamos clic en un contacto lanza la página ViewContact y nos muestra toda la información del contacto.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html
        xmlns:wicket="http://wicket.apache.org/dtds.data/wicket-xhtml1.4-strict.dtd">
<body>
        <wicket:extend>
                <div class="agendacontent">
                        <form wicket:id="deleteForm">
                                <table class="dataview">
                                        <tbody>
                                                <tr wicket:id="simple">
                                                        <td><a wicket:id="id"> <span
wicket:id="name"></span>
                                                        </a></td>
                                                </tr>
                                        </tbody>
                                </table>
                        </form>
```

```
                        <div id="fnav" wicket:id="navigator"></div>
                </div>
        </wicket:extend>
</body>
</html>
```

Listado 10. HTML de la clase FirstPage

Este es el archivo FirstPage.html que relaciona las posiciones de la página web con los objetos que utiliza y modifica la clase FirstPage.

```
package es.josanme.tool.pages;
import org.apache.wicket.markup.html.basic.Label;
import org.apache.wicket.markup.html.form.Button;
import org.apache.wicket.markup.html.form.Form;
import org.apache.wicket.markup.html.form.TextField;
import org.apache.wicket.markup.html.link.Link;
import org.apache.wicket.model.CompoundPropertyModel;
import es.josanme.model.Contacto;
import es.josanme.model.Mysql;

public class SecondPage extends BasePage {
        Link<Void> toThirdPageLink;
        String name;
        String phone;
        String email;
        String surName;
        public SecondPage() {
                disableLink(secondLink);
                initGui();
        }
        private void initGui() {
                Form<SecondPage> addLocationForm = new Form<SecondPage>(
                                "addLocationForm",                                      new
CompoundPropertyModel<SecondPage>(this));
                add(addLocationForm);
                Label nameLabel = new Label("nameLabel", "Name:  ");
                addLocationForm.add(nameLabel);
                TextField<String> nameField = new TextField<String>("name");
                addLocationForm.add(nameField);
                Label surNameLabel = new Label("surNameLabel", "Surname:");
                addLocationForm.add(surNameLabel);
                TextField<String> surNameField = new TextField<String>("surName");
                addLocationForm.add(surNameField);
                Label emailLabel = new Label("emailLabel", "Email:");
                addLocationForm.add(emailLabel);
                TextField<String> emailField = new TextField<String>("email");
                addLocationForm.add(emailField);
```

```
                Label phoneLabel = new Label("phoneLabel", "Phone:");
                addLocationForm.add(phoneLabel);
                TextField<String> phoneField = new TextField<String>("phone");
                addLocationForm.add(phoneField);
                Button submitButton = new Button("submitButton") {
                        @Override
                        public void onSubmit() {
                                Mysql bd = new Mysql();
                                Contacto nCon = new Contacto(name, surName, email, phone);
                                if (bd.newContact(nCon))
                                        info("Conctact added");
                                else
                                        error("Error this contact alredy exist");
                        }
                };
                addLocationForm.add(submitButton);
        }
}
```

Listado 11. Clase SecondPage

Esta es la clase SecondPage, es la que se encarga de insertar un contacto a la base de datos, esto se hace llamando al método newContact de mysql. Esta consta de 4 Labels, 4 TextField y un botón que es el que envía el contacto una vez hayamos rellenado los TextField.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html
        xmlns:wicket="http://wicket.apache.org/dtds.data/wicket-xhtml1.4-strict.dtd">
<body>
        <wicket:extend>
                <div class="agendacontent">
                        <form wicket:id="addLocationForm">
                                <table>
                                        <tr>
                                                <td></td>
                                                <td>
                                                        <h3>Add New Contact</h3>
                                                </td>
                                        </tr>
                                        <tr>
                                                <td></td>
                                        </tr>
```

```
                                        <tr>
                                                <td><label for="name"
wicket:id="nameLabel"></label></td>
                                                <td><input type="text" id="name"
wicket:id="name" /></td>
                                        </tr>

                                        <tr>
                                                <td><label for="surName"
wicket:id="surNameLabel"></label></td>
                                                <td><input type="text" id="surName"
wicket:id="surName" /></td>
                                        </tr>

                                        <tr>
                                                <td><label for="email"
wicket:id="emailLabel"></label></td>
                                                <td><input type="text" id="email"
wicket:id="email" /></td>
                                        </tr>

                                        <tr>
                                                <td><label for="phone"
wicket:id="phoneLabel"></label></td>
                                                <td><input type="text" id="phone"
wicket:id="phone" /></td>
                                        </tr>
                                        <tr></tr>
                                        <tr>
                                                <td></td>
                                                <td colspan="3" align="center"><input
type="submit"
                                                        value="Save" class="formbutton"
name="save"
                                                        wicket:id="submitButton" /></td>
                                        </tr>

                                </table>
                        </form>
                </div>
        </wicket:extend>
</body>
</html>
```

Listado 12. HTML de la clase SecondPage

Este es el archivo SecondPage.html que relaciona las posiciones de la página web con los objetos que utiliza y modifica la clase SecondPage.

```
package es.josanme.tool.pages;
import org.apache.wicket.ResourceReference;
import org.apache.wicket.markup.html.basic.Label;
import org.apache.wicket.markup.html.form.Form;
import org.apache.wicket.markup.html.form.ImageButton;
import org.apache.wicket.model.CompoundPropertyModel;

import es.josanme.model.Contacto;
import es.josanme.model.Mysql;

public class ViewContact extends BasePage {

	public ViewContact() {
		final Contacto cont = new Contacto("Adria","Sansaloni","godaking@gmail.com","629229493");
		Form<ViewContact> viewContact = new Form<ViewContact>("contact", new CompoundPropertyModel<ViewContact>(this));
		add(viewContact);

		viewContact.add(new Label("name",cont.getName()));

		viewContact.add(new Label("surName","surName: "));
		viewContact.add(new Label("surNameValue",cont.getSurName()));

		viewContact.add(new Label("email","Email: "));
		viewContact.add(new Label("emailValue",cont.getEmail()));

		viewContact.add(new Label("phone","Phone: "));
		viewContact.add(new Label("phoneValue",cont.getPhone()));

		viewContact.add(new ImageButton("delete",new ResourceReference(this.getClass(), "remove_256.png")) {
	    /** Delete from persistent storage, commit transaction. */
		  @Override
		public void onSubmit() {

	   Mysql bd = new Mysql();
	   bd.deleteContact(cont);
	   setResponsePage(new FirstPage());
	   info("delete");

		}
		});
	}
```

```
	public ViewContact(final Contacto cont)
	{
		Form<ViewContact> viewContact = new Form<ViewContact>("contact", new
CompoundPropertyModel<ViewContact>(this));
		add(viewContact);

		viewContact.add(new Label("name",cont.getName()));

		viewContact.add(new Label("surName","surName: "));
		viewContact.add(new Label("surNameValue",cont.getSurName()));

		viewContact.add(new Label("email","Email: "));
		viewContact.add(new Label("emailValue",cont.getEmail()));

		viewContact.add(new Label("phone","Phone: "));
		viewContact.add(new Label("phoneValue",cont.getPhone()));

		viewContact.add(new ImageButton("delete",new
ResourceReference(this.getClass(), "remove_256.png")) {
	    /** Delete from persistent storage, commit transaction. */
	       @Override
		public void onSubmit() {

	   Mysql bd = new Mysql();
	   bd.deleteContact(cont);
	   setResponsePage(new FirstPage());
	   info("delete");
	     }
	     });
	}
}
```

Listado 13. Clase ViewContact

Esta es la clase ViewContact que es la que se lanza al pulsar un contacto desde la lista de contactos. Costa de 7 labels cons los datos del contacto y un botón de eliminar que si lo pinchas eliminas el contacto.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html
	xmlns:wicket="http://wicket.apache.org/dtds.data/wicket-xhtml1.4-strict.dtd">
<body>
	<wicket:extend>
		<div class="agendacontent">
			<form wicket:id="contact">
				<table>
```

```
                                        <tr></tr>
                                        <tr>
                                                <td colspan="3" align="center">
                                                        <h1 wicket:id="name"></h1>
                                                </td>
                                        </tr>
                                        <tr>
                                                <td><label wicket:id="surName"></label></td>
                                                <td><p wicket:id="surNameValue"></p></td>
                                        </tr>
                                        <tr>
                                                <td><label wicket:id="email"></label></td>
                                                <td><p wicket:id="emailValue"></p></td>
                                        </tr>
                                        <tr>
                                                <td><label wicket:id="phone"></label></td>
                                                <td><p wicket:id="phoneValue"></p></td>
                                        </tr>
                                                <td colspan="3" align="center"><a><input
type="image"
                                                        wicket:id="delete" title="Delete
this task" src="remove_256.png"
                                                        onclick="return confirm('Are you
sure you want to delete this Contact?');document.location.reload(true);" />
                                                </a></td>
                                        </tr>
                                </table>
                        </form>
                </div>
        </wicket:extend>
</body>
</html>
```

Listado 14. HTML de la clase ViewContact

Este es el archivo ViewContact.html que relaciona las posiciones de la página web con los objetos que utiliza y modifica la clase ViewContact.

La aplicación tiene un funcionamiento muy sencillo. En la parte superior a la izquierda hay dos enlaces que son list y new Contact. El primero lleva a la FirstPage, que es la lista de contactos y es la que se ejecuta nada más empezar. El segundo enlace lleva a la SecondPage, es la página que introduce contactos a la lista.

A continuación pondré un par de capturas de la interfaz de la aplicación.

Figura 45: Lista de Contactos

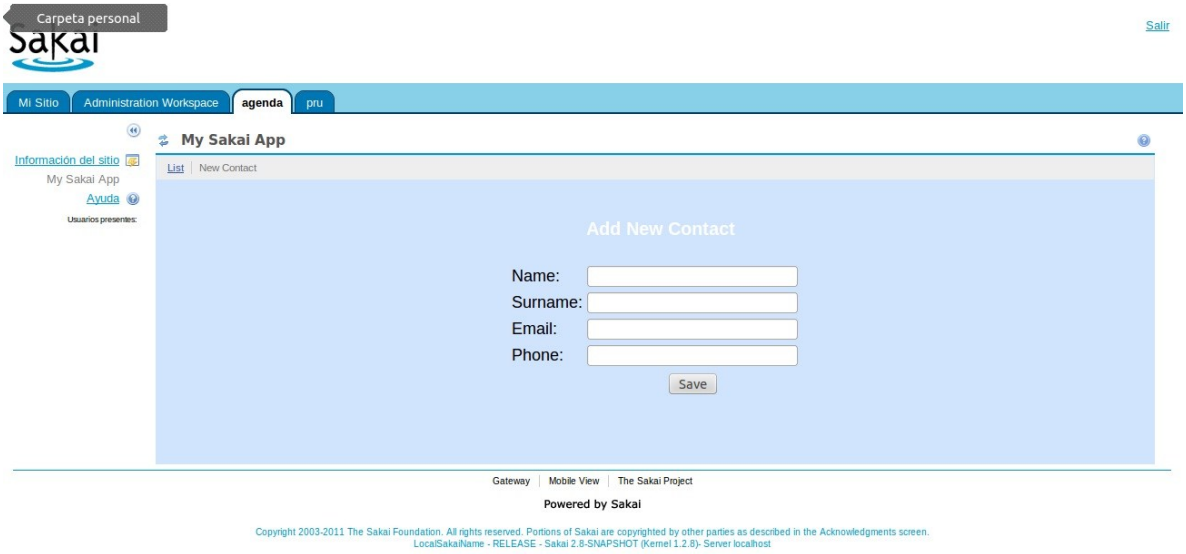

Figura 46: Nuevo Contacto

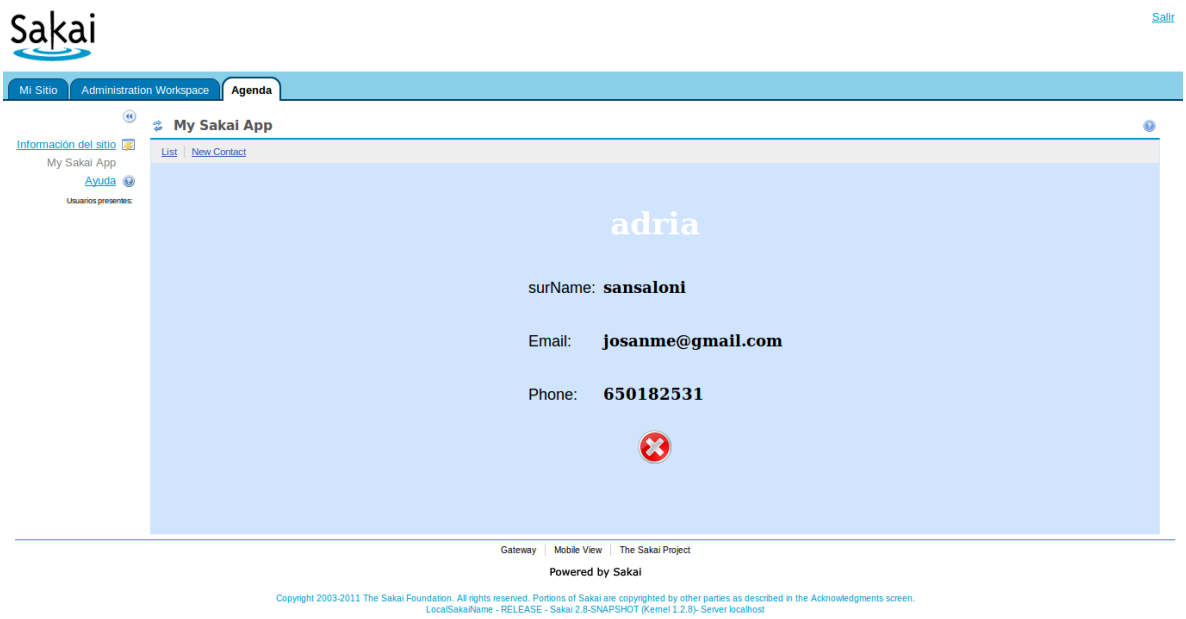

Figura 47: Eliminar Contacto

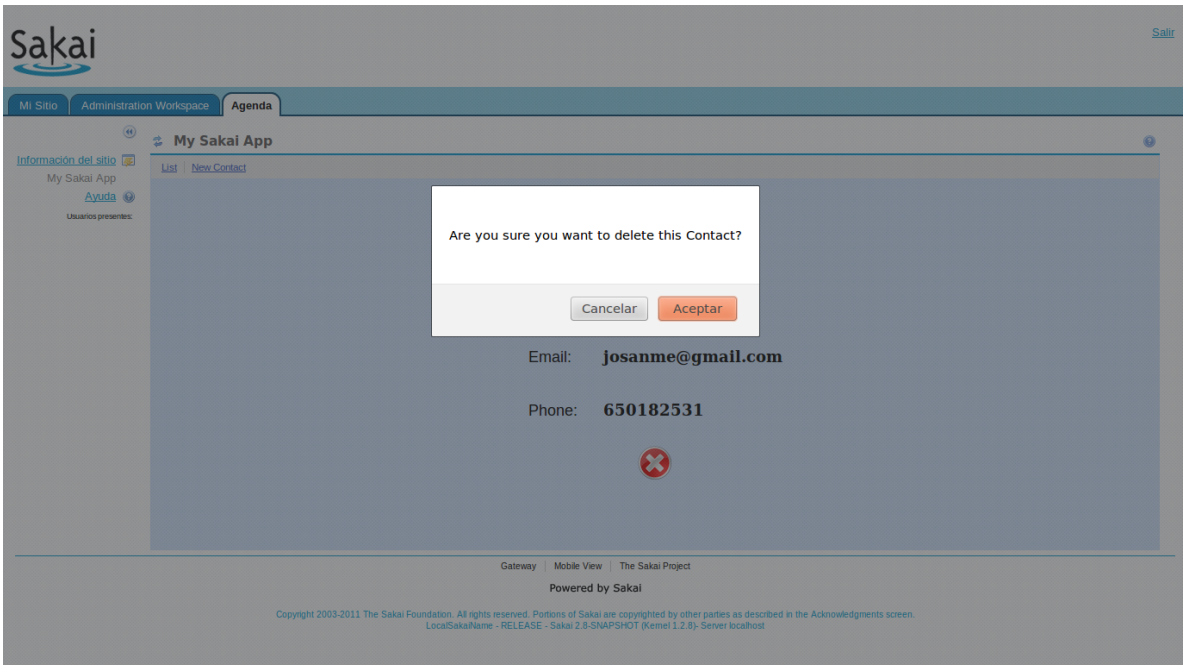

Figura 48: Validación de eliminar contacto

CAPITULO 7: DESARROLLO DE APLICACIÓN DE GESTOR DE TAREAS

La aplicación que hemos desarrollado consiste en un gestor de Tareas. A continuación explicaremos las clases que forman la aplicación desarrollada:

1. BasePage: Es la clase principal, heredada por todas las demás y se encarga de crear la base de las páginas, en este caso en concreto también crea dos links que redireccionan a la página que lista las tareas y a la página que crea Nuevas tareas. El código de esta clase lo podemos ver a continuación:

```
public class BasePage extends WebPage implements IHeaderContributor {

    private static final Logger log = Logger.getLogger(BasePage.class);

    @SpringBean(name="app1.logic.SakaiProxy")
    protected SakaiProxy sakaiProxy;

    @SpringBean(name="app1.logic.ProjectLogic")
    protected ProjectLogic projectLogic;

    Link<Void> firstLink;
    Link<Void> secondLink;
    final MySQL conector= new MySQL();
    ResultSet rs;
    String name="";
    ArrayList list = new ArrayList();
    FeedbackPanel feedbackPanel;
     public BasePage() {

            //first link
                firstLink = new Link<Void>("firstLink") {
                        private static final long serialVersionUID = 1L;
                        public void onClick() {
                                setResponsePage(new newTarea());
                        }
                };
                firstLink.add(new Label("firstLinkLabel",new
ResourceModel("link.first")).setRenderBodyOnly(true));
                firstLink.add(new AttributeModifier("title", true, new
ResourceModel("link.first.tooltip")));
                add(firstLink);

            //second link
                secondLink = new Link<Void>("secondLink") {
                        private static final long serialVersionUID = 1L;
                        public void onClick() {
                                setResponsePage(new Bpage());
                        }
                };
                secondLink.add(new Label("secondLinkLabel",new
```

```
ResourceModel("link.second")).setRenderBodyOnly(true));
                        secondLink.add(new AttributeModifier("title", true, new
ResourceModel("link.second.tooltip")));
                        add(secondLink);
        }
    public void clearFeedback(FeedbackPanel f) {
            if(!f.hasFeedbackMessage()) {
                    f.add(new SimpleAttributeModifier("class", ""));
            }
    }
    public void renderHead(IHeaderResponse response) {
            String skinRepo = sakaiProxy.getSkinRepoProperty();
            String toolCSS = sakaiProxy.getToolSkinCSS(skinRepo);
            String toolBaseCSS = skinRepo + "/tool_base.css";
            response.renderJavascriptReference("/library/js/headscripts.js");
            response.renderCSSReference(toolBaseCSS);
            response.renderCSSReference(toolCSS);
    response.renderOnLoadJavascript("setMainFrameHeight( window.name )");
    response.renderString("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\"
/>");
    }
    protected void disableLink(Link<Void> l) {
            l.add(new AttributeAppender("class", new Model<String>("current"), " "));
            l.setRenderBodyOnly(true);
            l.setEnabled(false);
    }


}
```

Listado 15. Clase BasePage

Este sería el archivo BasePage.html asociado al BasePage.java que se encarga de crear la página:

```
<html>
<head>
<title> Gestor de Tareas</title>
</head>

<body>
	<div class="portletBody">
		<ul class="navIntraTool actionToolbar" role="menu">
			<li class="firstToolBarItem" role="menuitem"><span> <a
					wicket:id="firstLink"> <span
wicket:id="firstLinkLabel">[firstLinkLabel]</span>
				</a>
			</span></li>
			<li class="secondToolBarItem" role="menuitem"><span> <a
					wicket:id="secondLink"> <span
wicket:id="secondLinkLabel">[secondLinkLabel]</span>
				</a>
			</span></li>
		</ul>


		<!--	<span wicket:id="feedback">feedbackmessages will be put here</span>-->

		<wicket:child />

	</div>
	<!--<span wicket:id="searchPanel"></span> -->
</body>
</html>
```

Listado 16. HTML de la clase BasePage

2. Tarea.java: Esta es una de las clases principales porque pese a no representar ninguna página es el objeto en el que se basa la aplicación. Tiene un código bastante simple con un constructor vacío y varios métodos consultores y modificadores para tratar con sus atributos. A continuación lo podemos ver:

```
public class Tarea implements Serializable{

  private Long id;
  private String fecha;
  private String hora;
  private String titulo;
```

```
    private String tarea;
    private String grupo;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getFecha() {
        return fecha;
    }

    public void setFecha(String fecha) {
        this.fecha = fecha;
    }

    public String getHora() {
        return hora;
    }

    public void setHora(String hora) {
        this.hora = hora;
    }

    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public String getTarea() {
        return tarea;
    }

    public void setTarea(String tarea) {
        this.tarea = tarea;
    }

    public String getGrupo() {
        return grupo;
    }

    public void setGrupo(String grupo) {
        this.grupo = grupo;
    }
}
```

Listado 17. Clase Tarea

3. MySQL.java: Esta clase es la que utlizamos para hacer de conexión con la base de datos. Tiene varios métodos para ello como, serian DeleteTarea(); que sirve para borrar una Tarea, BuscarTarea(); que lo utilizamos para buscar una tarea, list(); que devuelve una lista con todas las tareas y NuevaTarea() que crea una nueva tarea en la base de datos. El código de esta clase es:

```
public class MySQL {
	Connection conn = null;
	Statement stmt = null;
	ResultSet rs = null;

	public void con() {
		 {
		try {
conn =DriverManager.getConnection("jdbc:mysql://localhost/sakai?user=sakai&password=a");
			stmt = conn.createStatement();
			stmt.executeUpdate("CREATE TABLE IF NOT EXISTS agenda (id int not null
auto_increment,primary key(id),name varchar(20),surname varchar(50),email varchar(50),tlf varchar(15))");
			} catch (SQLException ex) {}
			}
			}
	public void close() {
		try {
			if (conn != null)
				conn.close();

			if (rs != null)
				rs.close();

			if (stmt != null)
				stmt.close();
		} catch (SQLException sqlEx) {
		}
	}

	public ResultSet list() {
		con();
		try {
			rs = stmt.executeQuery("select * from app1");
		} catch (SQLException e) {
		}
		return rs;
	}

	public boolean nuevaTarea(Tarea tar) {
		int numRow = -1;
		boolean state = false;
		con();
		try {
numRow = stmt.executeUpdate("insert into app1(Fecha, Hora, Titulo, Tarea, Relevancia) values('"+
tar.getFecha() + "','" + tar.getHora() + "','" + tar.getTitulo() + "','" + tar.getTarea() + "','" + tar.getGrupo() + "')");
```

```
                stmt.executeUpdate("commit");
                } catch (SQLException e) {}
                close();
                if (numRow > 0)
                        state = true;
                return state;
        }

        public boolean deleteTarea(Tarea t) {
                int numRow = -1;
                boolean state = false;
                long id=t.getId();
                int ids=(int) id;
                con();
                try {
                        numRow = stmt.executeUpdate("delete from app1 where id="
                                        + ids);
                        stmt.executeUpdate("commit");
                } catch (SQLException e) {
                }
                close();
                if (numRow > 0)
                        state = true;
                return state;

        }


        public Tarea buscarTarea(long id){
                int ids =(int) id;
                int numRow = -1;
                Tarea t =new Tarea();
                con();try {
                        rs.beforeFirst();
                        rs.next();
                        rs = stmt.executeQuery("select * from app1 where id='"
                                        + ids + "'");
                        stmt.executeUpdate("commit");
                        t.setFecha(rs.getString(2));
                        t.setHora(rs.getString(3));
                        t.setTitulo(rs.getString(4));
                        t.setTarea(rs.getString(5));
                        t.setGrupo(rs.getString(6));
                } catch (SQLException e) {
                }
                close();

                return t;

        }

}
```

Listado 18. Clase MySQL

MyApplication.java: Esta clase es la primera en ejecutarse al lanzar la aplicación de sakai, a partir de esta clase se desarrolla toda la aplicación. Su código es el siguiente:

```
public class MyApplication extends WebApplication {

	protected void init() {

		addComponentInstantiationListener(new SpringComponentInjector(this));

		getResourceSettings().setThrowExceptionOnMissingResource(false);

		getMarkupSettings().setStripWicketTags(true);

		getMarkupSettings().setDefaultBeforeDisabledLink(null);
		getMarkupSettings().setDefaultAfterDisabledLink(null);

		getApplicationSettings().setPageExpiredErrorPage(Bpage.class);
		getApplicationSettings().setAccessDeniedPage(Bpage.class);


	}



	public RequestCycle newRequestCycle(Request request, Response response) {
		return new WebRequestCycle(this, (WebRequest)request, (WebResponse)response) {
			public Page onRuntimeException(Page page, RuntimeException e) {
				throw e;
			}
		};
	}


	public Class <Bpage> getHomePage() {
		return Bpage.class;
	}

	public MyApplication()
	{
	}

}
```

Listado 19. Clase MyApplication

Esta clase viene apoyada por el archivo MyApplication.properties que es donde se definen algunas de las propiedades principales de la aplicación. su código viene a continuación:

```
page.title = Lista Tareas

link.first = Nueva Tarea
```

```
link.first.tooltip = Go to the home page

link.second = Mis Tareas
link.second.tooltip = This is another page

# An example of parameter substitution
link.third = Page Number {0}
link.third.tooltip = Yet another page

the.time = Today is {0} and the time is currently {1}

goto.page.three = Go to page three
```

Listado 20. Archivo MyApplication.properties

Bpage.java: En esta clase es donde se muestra la lista de todas las tareas que hay en la base de datos. Este es su código:

```
public class Bpage extends BasePage {
	Tarea p=new Tarea();
	public Bpage(){
		//p.setTitulo("Peta");
		ResultSet rs;
		ArrayList list = new ArrayList();
		final MySQL conector= new MySQL();
		Tarea Tarea=null;
		try{
	rs = conector.list();
	rs.beforeFirst();
	while(rs.next()){
		Tarea=new Tarea();
	Tarea.setFecha(rs.getString(2));
	Tarea.setHora(rs.getString(3));
	Tarea.setTitulo((String)rs.getString(4));
	Tarea.setTarea((String)rs.getString(5));
	Tarea.setGrupo((String)rs.getString(6));
	list.add(Tarea);

	}
    conector.close();
	}catch(SQLException e){}

		final Form<Bpage> addDeleteForm = new Form<Bpage>("deleteForm",
		new CompoundPropertyModel<Bpage>(this));
		add(addDeleteForm);
		final DataView dataView = new DataView("simple", new ListDataProvider(
				list)) {
			public void populateItem(final Item item) {

				Link view = new Link("id", item.getModel()) {
	public void onClick() {
```

```
                p = (Tarea) getModelObject();
                setResponsePage(new Vista(p));
            }
        };

        view.add(new Label("message",p.getTitulo()));
        Tarea t = (Tarea)item.getModelObject();
        view.add(new Label("name",t.getTitulo()));
        item.add(view);


                        }

                };

                dataView.setItemsPerPage(5);

                addDeleteForm.add(dataView);

                add(new PagingNavigator("navigator", dataView));

        }

}
```

Listado 221. Clase BPage

Esta clase está representada por el Bpage.html, que su código es:

```
<wicket:extend>
<h1>
        Mis Tareas
</h1>
<form wicket:id="deleteForm">
                        <table cellspacing="0" class="dataview">
                                <tbody>
                                        <tr wicket:id="simple">
                                                <td><a wicket:id="id">
                                                        <span wicket:id="message"></span>
                                                        <span wicket:id="name"></span>
                                                </a></td>

                                        </tr>
                                        <tr></tr>
                                        <tr></tr>
                                        <tr></tr>
                                        <tr></tr>
                                </tbody>
                        </table>
```

```
        </form>
        <div wicket:id="navigator"></div>

</wicket:extend>
```

Listado 22. HTML de la clase BasePage

Y esta clase nos da la representación de la siguiente página:

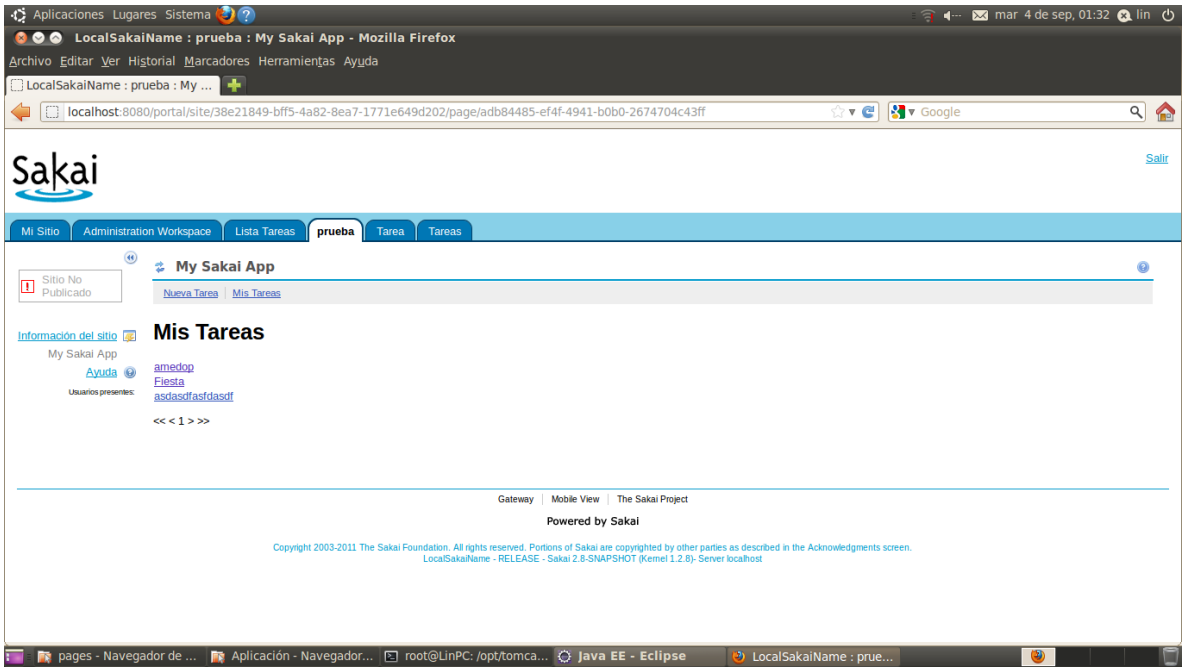

Figura 49: Lista de Tareas

NewTarea.java: En esta clase nos encargamos de anyadir las nuevas tareas a la base de datos. Su código es el siguiente:

```
public class NewTarea extends BasePage {
          final MySQL conector= new MySQL();

          public NewTarea() {

          setDefaultModel(new CompoundPropertyModel(new LoadableDetachableModel() {
        protected Object load() {
          return new Tarea();
        }
    }));
    init();
  }

  public NewTarea (final Long contactId) {
    setDefaultModel(new CompoundPropertyModel(new LoadableDetachableModel() {
      protected Object load() {return 0;
      }
    }));
                    MySQL conec=new MySQL();
    init();
  }

  private void init() {
    add(new FeedbackPanel("feedback"));
    add(new TareaForm("form", getDefaultModel()));
  }

  private class TareaForm extends Form {

    public TareaForm(String id, IModel m) {
      super(id, m);

      TextField fecha = new TextField("fecha");
      fecha.setRequired(true);
      fecha.add(StringValidator.maximumLength(15));
      add(fecha);

      TextField hora = new TextField("hora");
      hora.setRequired(true);
      hora.add(StringValidator.maximumLength(20));
      add(hora);

      TextField titulo = new TextField("titulo");
      titulo.add(StringValidator.maximumLength(50));
      add(titulo);

      TextArea tarea = new TextArea("tarea");
      tarea.add(StringValidator.maximumLength(500));
      add(tarea);
```

```
        DropDownChoice grupo = new DropDownChoice("grupo");
        grupo.setChoices(new AbstractReadOnlyModel() {
           public Object getObject() {
              List<String> l = new ArrayList<String>(3);
              l.add("Imprescindible");
              l.add("Normal");
              l.add("Irrelevante");
              return l;
           }
        });
        add(grupo);

        add(new Button("save") {
           public void onSubmit() {
              Tarea t = (Tarea) getForm().getModelObject();
              conector.nuevaTarea(t);

               setResponsePage(Bpage.class);
           }
        });
        add(new Button("cancel") {
           public void onSubmit() {
              setResponsePage(Bpage.class);
           }
        }.setDefaultFormProcessing(false));
     }
  }
}
```

Listado 23. Clase NewTarea

Esta clase viene representada por NewTarea.html, que tiene el siguiente código:

```
<wicket:extend>
<h1>
          Nueva Tarea
</h1>
  <span wicket:id="feedback"></span>

<!-- <span wicket:id="message"></span> -->

  <form wicket:id="form">
    <div id="contacts">
      <div class="tarea">
        <table>
          <tr>
            <td>
              Fecha
            </td>
            <td>
```

```
                <input wicket:id="fecha" type="text"/>
              </td>
            </tr>
            <tr>
              <td>
                Hora
              </td>
              <td>
                <input wicket:id="hora" type="text"/>
              </td>
            </tr>
            <tr>
              <td>
                Titulo
              </td>
              <td>
                <input wicket:id="titulo" type="text" size="40"/>
              </td>
            </tr>
            <tr>
              <td>
                Tarea
              </td>
              <td>
                <textarea wicket:id="tarea" rows="4" cols="40"></textarea>
              </td>
            </tr>
            <tr>
              <td>
                Relevancia
              </td>
              <td>
                <select wicket:id="grupo"> </select>
              </td>
            </tr>
            <tr>
              <td colspan="3" align="center">
                <input wicket:id="save" type="submit" value="Save"/>
                <input wicket:id="cancel" type="submit" value="Cancel"/>
              </td>
            </tr>
          </table>
        </div>
      </div>
    </form>
</wicket:extend>
```

Listado 24. HTML de la clase NewTarea

Esta clase nos proporciona la siguiente página:

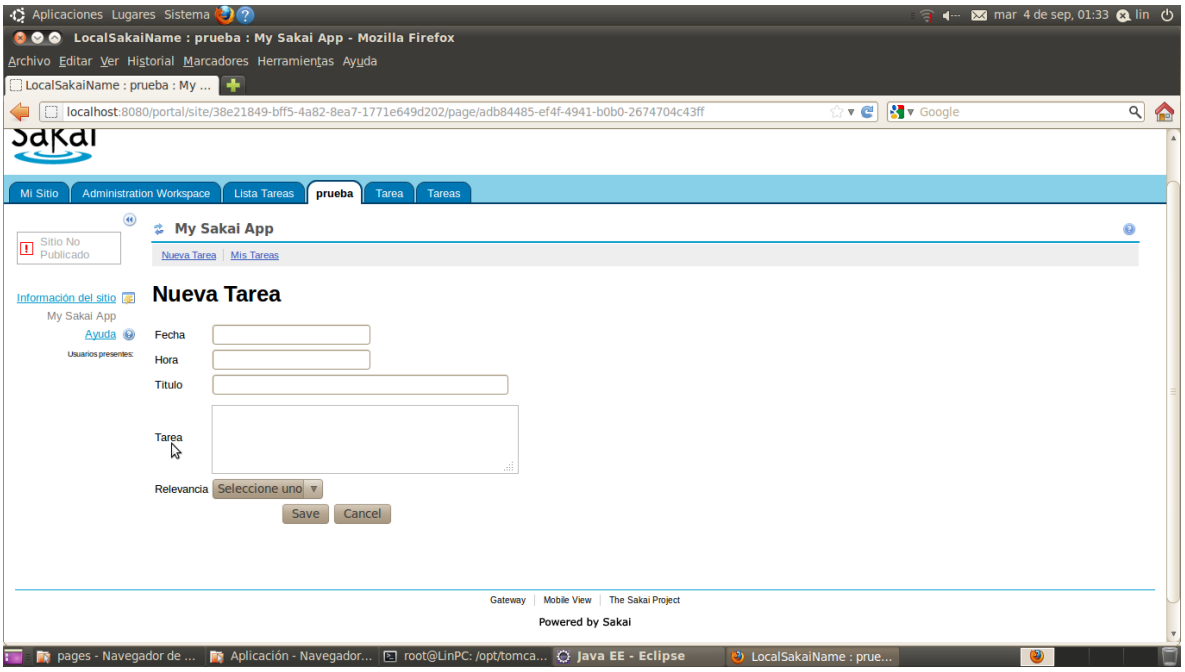

Figura 50: Nueva Tarea

Vista: Esta clase muestra toda la información de la tarea con detalle. Se ejecuta cuando pinchas encima de una tarea de los que aparece en la lista. Cuando estas en esta página hay un botón que si pinchas sobre el elimina la tarea. El código de esta clase es:

```
public class Vista extends BasePage{

	public Vista(final Tarea t){
		final MySQL cont = new MySQL();

		Form<Vista> vista = new Form<Vista>("vistaT", new
CompoundPropertyModel<Vista>(this));
		add(vista);
		vista.add(new Label("titulo",t.getTitulo()));
		vista.add(new Label("Nfecha","Fecha: "));
		vista.add(new Label("fecha",t.getFecha()));
		vista.add(new Label("Nhora","Hora: "));
		vista.add(new Label("hora",t.getHora()));
		vista.add(new Label("Ntarea","Tarea: "));
		vista.add(new Label("tarea",t.getTarea()));
		vista.add(new Label("Ngrupo","Relevancia: "));
		vista.add(new Label("grupo",t.getGrupo()));


		vista.add(new ImageButton("delete",new
ResourceReference(this.getClass(), "delete_big.png")) {
		/** Delete from persistent storage, commit transaction. */
```

```
            @Override
           public void onSubmit() {

                 //long a=t.getId();
           //     int b= (int)a;
        //cont.deleteTarea(t);
        setResponsePage(new Bpage());
        info("delete");


          }
          });


           }
}
```

Listado 25. Clase Vista

El archivo html que se encarga de mostrar esta clase como página es Vista.html, el cual podemos ver a continuación:

```
<wicket:extend>
<form wicket:id="vistaT">
<h1><span wicket:id="titulo"></span></h1>

<table>
                        <tr>
                              <td><label wicket:id="Nfecha"></label></td>
                              <td><span wicket:id="fecha"></span></td>
                        </tr>

                        <tr>
                              <td><label wicket:id="Nhora"></label></td>
                              <td><span wicket:id="hora"></span></td>
                        </tr>



                        <tr>
                              <td><label wicket:id="Ntarea"></label></td>
                              <td><span wicket:id="tarea"></span></td>
                        </tr>
                         <tr>
                              <td><label wicket:id="Ngrupo"></label></td>
                              <td><span wicket:id="grupo"></span></td>
                        </tr>

                        <tr></tr>
                        <tr></tr>
```

```
                        <tr>
                              <td colspan="3" align="center">
                                    <a><input type="image" wicket:id="delete"
                                    title="Eliminar esta tarea"
src="delete_big.png"
                                    onclick="return confirm('Estas seguro de
que quieres eliminar esta tarea?');document.location.reload(true);" />
                                    </a>

                              </td>
                        </tr>


</table>
</form>
</wicket:extend>
```

Listado 26. HTML de la clase Vista

Y un ejemplo que muestra la vista de una tarea lo podemos ver en la siguiente imagen:

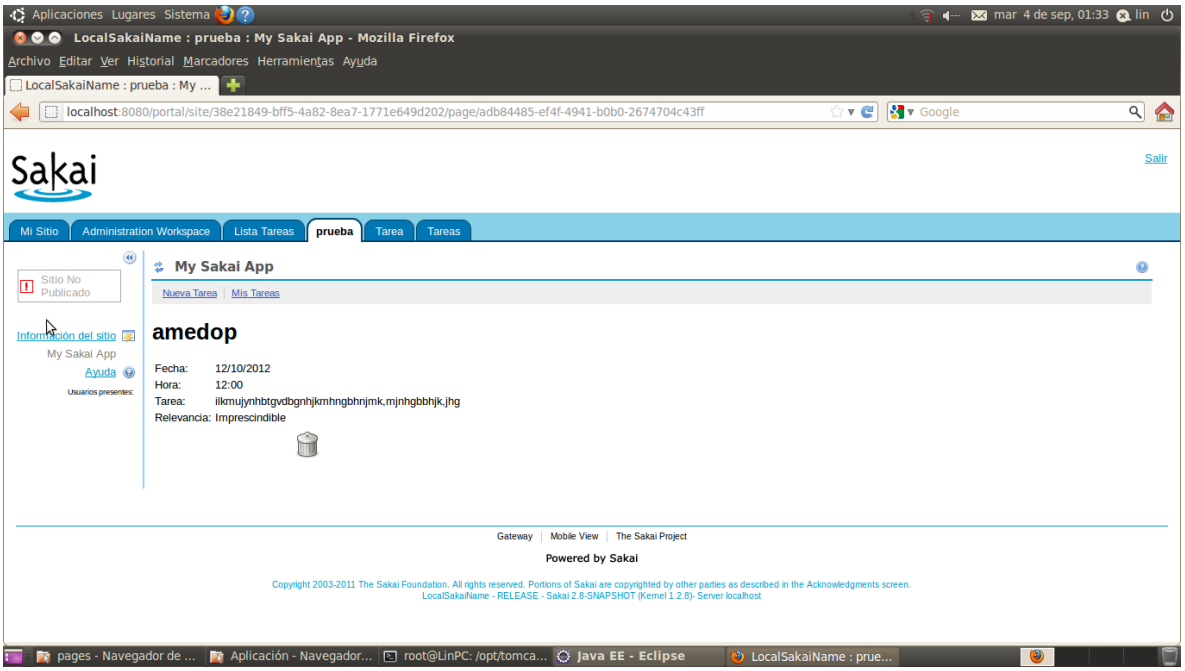

Figura 51: Ver Tarea

La aplicación tiene un funcionamiento muy simple. En la parte superior a la izquierda hay dos enlaces que son Mis Tareas y Nueva Tarea. El primero lleva a la Bpage, que es la lista de tareas y es la que se ejecuta nada más empezar. El segundo enlace lleva a la NewTarea, es la página que introduce tareas a la lista. Si quieres ver una Tarea de forma más detallada solo tienes que clicar sobre una de ellas.

CAPITULO 8: PRUEBAS

Durante el proyecto se han dessarrollado dos aplicaciones de sakai y dos scripts que automatizan la instalación de la plataforma Sakai. Los scripts de instalación han sido provadas en tres maquinas distintas todas conectadas a intenet mediante conexión por cable. Las aplicaciones han sido provadas en estas instalaciones de la plataforma de Sakai y tanto la Agenda de contactos, como el gestor de Tareas, han funcionado perfectamente.

CAPITULO 9: CONCLUSIÓN

Para finalizar el proyecto podemos confirmar que nos hemos introducido mínimamente en la plataforma Sakai y hemos aprendido a trabajar con Maven, Tomcat y MySQL. Hemos concluido la instalación de Sakai, creado un script para su automatización y desarrollado una aplicación dentro de su plataforma. Como pasos para realizar en el futuro se debería probar a instalar la versión ***trunk*** de Sakai. Finalmente el script, el material necesario para el script y la aplicación desarrollada se adjuntaran junto con este documento en un archivo comprimido.

Recomendaciones:

Se recomienda que a la hora de ejecutar los scripts este conectado a internet con conexión por cable, ya que, nosotros hemos observado que si estas conectado a internet a través de wifi, da errores a la hora de descargar los archivos binarios de sakai mediante subversión.

CAPITULO 10: BIBLIOGRAFIA

MySQL 5.5 Manual, "Reference Manual"
Agosto 2012. [Online]. Disponible:
http://dev.mysql.com/doc/refman/5.0/es/introduction.html

Apache Maven Project, "Introduction to the POM"
Agosto 2012. [Online]. Available:
http://maven.apache.org/guides/introduction/introduction-to-the-pom.html

Apache Maven Project, "Introduction"
Agosto 2012. [Online]. Disponible:
http://maven.apache.org/what-is-maven.html

Wikipedia, "Maven"
Enero 2012. [Online]. Disponible:
http://es.wikipedia.org/wiki/Maven#Ejemplos

Tigris.org, "Subeclipse FAQ"
Noviembre 2009. [Online]. Disponible:
http://subclipse.tigris.org/wiki/PluginFAQ

The Apache Software foundation, "The Apache Tomcat 5.5 Servlet/JSP Container"
[Online]. Disponible:
Sakai

Wikipedia, "Tomcat"
Agosto 2012. [Online]. Disponible:
http://es.wikipedia.org/wiki/Tomcat

Oracle "Java Servlet Technology Overview"
[Online]. Disponible:
http://www.oracle.com/technetwork/java/javaee/servlet/index.html

Programación en Castellano, Alberto Cardona "Servlets y JSP"
[Online]. Disponible:
http://www.programacion.com/articulo/servlets_y_jsp_82

Wikipedia, "Java Servlet"
Agosto 2012. [Online]. Disponible:
http://es.wikipedia.org/wiki/Java_Servlet

Geneura, Juan Julian Merelo Guervos, “Programando con JSPs ”
Octubre 2004. [Online]. Disponible:
http://geneura.ugr.es/~jmerelo/JSP/

Wikipedia, “JavaServer Pages”
Julio 2012. [Online]. Disponible:
http://es.wikipedia.org/wiki/JavaServer_Pages

Sakai Project,”Sakai”
[Online]. Disponible:
http://www.Sakaiproject.org